

#2  
Priority  
Paper  
8-11-01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of	)	
Masahito MATSUO	)	Group Art Unit: Unassigned
Application No.: Unassigned	)	Examiner: Unassigned
Filed: August 6, 2001	)	
For: DATA PROCESSOR	)	
	)	
	)	
	)	
	)	
	)	
	)	

Jc971 U.S. PTO  
09/921554  
08/06/01

CLAIM FOR CONVENTION PRIORITY

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

The benefit of the filing date of the following prior foreign application in the following foreign country is hereby requested, and the right of priority provided in 35 U.S.C. § 119 is hereby claimed:

Japanese Patent Application No. 2001-027416

Filed: February 2, 2001

In support of this claim, enclosed is a certified copy of said prior foreign application. Said prior foreign application was referred to in the oath or declaration. Acknowledgment of receipt of the certified copy is requested.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

Date: August 6, 2001

By: \_\_\_\_\_

Platon N. Mandros  
Registration No. 22,124

P.O. Box 1404  
Alexandria, Virginia 22313-1404  
(703) 836-6620

日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application:

2001年 2月 2日

出 願 番 号  
Application Number:

特願2001-027416

出 願 人  
Applicant (s):

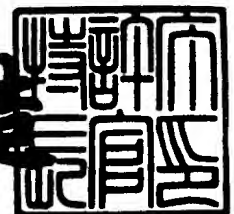
三菱電機株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年 2月23日

特許庁長官  
Commissioner,  
Patent Office

及 川 耕 造



PATENT OFFICE  
JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of  
the following application as filed with this Office.

Date of Application : February 2, 2001

Application Number : Japanese Patent Application No. 2001-027416

Applicant(s) : MITSUBISHI DENKI KABUSHIKI KAISHA

This 23rd day of February, 2001

Commissioner,  
Patent Office Kozo OIKAWA

Certificate No. 2001-3010541

【書類名】 特許願

【整理番号】 528307JP01

【提出日】 平成13年 2月 2日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/38  
G06F 9/32

【発明者】

【住所又は居所】 東京都千代田区丸の内二丁目 2 番 3 号 三菱電機株式会  
社内

【氏名】 松尾 雅仁

【特許出願人】

【識別番号】 000006013

【氏名又は名称】 三菱電機株式会社

【代理人】

【識別番号】 100066474

【弁理士】

【氏名又は名称】 田澤 博昭

【選任した代理人】

【識別番号】 100088605

【弁理士】

【氏名又は名称】 加藤 公延

【手数料の表示】

【予納台帳番号】 020640

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要



【書類名】 明細書

【発明の名称】 情報処理装置

【特許請求の範囲】

【請求項1】 複数の命令からなり繰り返し処理されるリピートブロックを含むプログラムを実行する情報処理装置において、

実行される命令で特定されるオペレーションとは独立して、上記リピートブロック内で繰り返し処理の終了を検出する、ハードウェアにより実現された検出手段と、

上記検出手段が繰り返し処理の終了を検出した場合、上記リピートブロック内の残りの命令の実行を抑止する命令実行抑止手段と

を備えたことを特徴とする情報処理装置。

【請求項2】 命令実行抑止手段は、検出手段が繰り返し処理の終了を検出した場合、リピートブロック内の残りの命令をノーオペレーション化する、ハードウェアにより実現された手段であることを特徴とする請求項1記載の情報処理装置。

【請求項3】 命令実行抑止手段は、検出手段が繰り返し処理の終了を検出した場合、命令フェッチ段階でリピートブロックの次の命令に命令処理シーケンスを切り替える命令処理シーケンス切替手段であることを特徴とする請求項1記載の情報処理装置。

【請求項4】 命令実行抑止手段は、検出手段が繰り返し処理の終了を検出した場合、命令実行段階でリピートブロックの次の命令に命令処理シーケンスを切り替える命令処理シーケンス切替手段であることを特徴とする請求項1記載の情報処理装置。

【請求項5】 命令処理シーケンス切替手段は、リピートブロック内の繰り返し処理において最後に実行される最終命令を実行時に、リピートブロックの次の命令へのジャンプ処理を行う手段であることを特徴とする請求項4記載の情報処理装置。

【請求項6】 命令処理シーケンス切替手段は、リピートブロック内の繰り返し処理において最後に実行される最終命令を実行後に、リピートブロックの次

の命令へのジャンプ処理を行う手段であることを特徴とする請求項4記載の情報処理装置。

【請求項7】 検出手段は、リピートブロックの繰り返し処理時に実行される命令のアドレスに基づき繰り返し処理が終了したか否かを判定する手段であることを特徴とする請求項1から請求項6のうちのいずれか1項記載の情報処理装置。

【請求項8】 検出手段は、リピートブロックを繰り返し処理する回数をカウントするカウント手段と、上記リピートブロック内の実行される命令のアドレスと最後に実行される最終命令のアドレスとを比較する比較手段とを有しており、上記カウント手段のカウント数が所定の数に到達し、上記比較手段が実行される命令のアドレスが最終命令のアドレスと一致したことを通知した際に繰り返し処理の終了と判定することを特徴とする請求項7記載の情報処理装置。

【請求項9】 検出手段は、リピートブロックの繰り返し処理時に実行される命令数に基づき繰り返し処理が終了したか否かを判定する手段であることを特徴とする請求項1から請求項6のうちのいずれか1項記載の情報処理装置。

【請求項10】 検出手段は、リピートブロック内の実行された命令数をカウントするカウント手段を有しており、上記カウント手段のカウント数が所定の数に到達した際に繰り返し処理の終了と判定することを特徴とする請求項9記載の情報処理装置。

【請求項11】 検出手段は、リピートブロックを繰り返し処理する回数をカウントする第1カウント手段と、上記リピートブロック内の実行された命令数をカウントする第2カウント手段とを有しており、上記第1カウント手段のカウント数が第1の所定の数に到達し、上記リピートブロックの最終の繰り返し処理において上記第2カウント手段のカウント数が第2の所定の数に到達した際に繰り返し処理の終了と判定することを特徴とする請求項9記載の情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、複数の命令からなり繰り返し処理されるリピートブロックを含む

プログラムを実行する情報処理装置に関するものである。

【0002】

【従来の技術】

一般に、デジタル信号処理は多くの繰り返し処理を含む。デジタル・シグナル・プロセッサ(DSP)は、デジタル信号処理を高速に実現する専用のプロセッサであり、単一命令や繰り返し処理(リピート処理)される複数の命令を含むリピートブロックを効率よく処理するためのリピート命令を備えているものが多い。

【0003】

一方、VLIW (Very Long Instruction Word) 技術を用いデジタル信号処理を効率よく行うマルチメディア向け情報処理装置が開発されている。図36は特開平9-212361号公報(米国特許5,901,301)に開示されたこのような従来の情報処理装置の一例のソフトウェアにより実現された繰り返し処理を示すフローチャート図である。特開平9-212361号公報に開示された従来の情報処理装置は、信号処理の流れをハードウェア化したDSPとは異なり、繰り返し処理を高速化するために、ロードのレイテンシ、レジスタ値のライフタイム等のソフトウェアパイプラインが必要であり、ある程度ソフトウェアで繰り返し処理を展開し最適化を行う必要がある。また、単純な積和演算処理でもメモリ上の2カ所の領域からデータを読み出す必要がある。それ故、高速な処理を実現するために、複数データの処理を繰り返し(ループの)単位としている。

【0004】

次に動作について説明する。

図36に示した例は、積和演算の基本処理単位が4回の処理でループ処理が実現できる場合であり、 $1, 2, \dots, 7, 4n, 4n+1, 4n+2, 4n+3$  ( $n$ は2以上の整数)回それぞれ積和演算処理を行う(すなわちリピートブロックを実行する)複数のプログラムが独立に設けられたものである。情報処理装置はまず、ステップST1において、リピートブロックを繰り返し処理する回数すなわち繰り返し回数が8以上であるか否かを判定する。繰り返し回数が8以上であ

る場合には、情報処理装置は、ステップST2において、繰り返し回数が $4n$ 、 $4n+1$ 、 $4n+2$ 、 $4n+3$ のいずれに等しいか否かをさらに判定し、その判定結果にしたがって積和演算処理を行うプログラムへの分岐を行い、そのプログラムを実行する（ステップST3a～ST3d）。一方、繰り返し回数が8未満である場合には、情報処理装置は、ステップST4において、繰り返し回数が1～7のいずれに等しいか否かをさらに判定し、その判定結果にしたがって積和演算処理を行うプログラムへの分岐を行い、そのプログラムを実行する（ステップST5a～ST5g）。

#### 【0005】

以上のように、リピートブロックを処理する繰り返し回数がダイナミックに変化する場合、情報処理装置はソフトウェアにより繰り返し回数を判定し繰り返し回数に対応したプログラムへ分岐する。

#### 【0006】

##### 【発明が解決しようとする課題】

従来の情報処理装置は以上のように構成されているので、リピートブロックを処理する繰り返し回数がダイナミックに変化する場合や、任意の繰り返し回数で同一サブルーチンを呼び出そうとする場合には、繰り返し回数を条件判定するためのオーバーヘッドが大きくなり、高い性能が得られないという課題があった。また、繰り返し回数の条件判定、この判定結果に基づく分岐、および繰り返し回数に応じた繰り返し処理のためのコードが必要となるため、繰り返し処理を実現するためのプログラムサイズが大きくなるという課題もあった。特に、ソフトウェアをROM化する場合には、実装ROMサイズはコードサイズに依存するのでハードウェアコストも高くなり、さらに、高速処理を行うために単純な繰り返し処理でも結構複雑なプログラムとなるので、プログラム開発負荷が大きくバグ混入の可能性も高いという課題もあった。

#### 【0007】

この発明は上記のような課題を解決するためになされたもので、1つのプログラムでリピートブロックを任意の繰り返し回数処理してリピートブロックの途中でリピート処理を終了させることを可能にし、それ故プログラムサイズを大幅に

削減するとともにプログラムの開発効率を向上させることができる、高性能で低コストな情報処理装置を得ることを目的とする。

【0008】

【課題を解決するための手段】

この発明に係る情報処理装置は、実行される命令で特定されるオペレーションとは独立して、リピートブロック内で繰り返し処理の終了を検出する、ハードウェアにより実現された検出手段と、上記検出手段が繰り返し処理の終了を検出した場合、上記リピートブロック内の残りの命令の実行を抑止する命令実行抑止手段とを備えたものである。

【0009】

この発明に係る情報処理装置は、命令実行抑止手段が、検出手段が繰り返し処理の終了を検出した場合、リピートブロック内の残りの命令をノーオペレーション化する、ハードウェアにより実現された手段であるものである。

【0010】

この発明に係る情報処理装置は、命令実行抑止手段が、検出手段が繰り返し処理の終了を検出した場合、命令フェッチ段階でリピートブロックの次の命令に命令処理シーケンスを切り替える命令処理シーケンス切替手段であるものである。

【0011】

この発明に係る情報処理装置は、命令実行抑止手段が、検出手段が繰り返し処理の終了を検出した場合、命令実行段階でリピートブロックの次の命令に命令処理シーケンスを切り替える命令処理シーケンス切替手段であるものである。

【0012】

この発明に係る情報処理装置は、命令処理シーケンス切替手段が、リピートブロックの繰り返し処理において最後に実行される最終命令の実行時に、リピートブロックの次の命令へのジャンプ処理を行う手段であるものである。

【0013】

この発明に係る情報処理装置は、命令処理シーケンス切替手段が、リピートブロックの繰り返し処理において最後に実行される最終命令の実行後に、リピートブロックの次の命令へのジャンプ処理を行うものである。

## 【0014】

この発明に係る情報処理装置は、検出手段が、リピートブロックの繰り返し処理時に実行される命令のアドレスに基づき繰り返し処理が終了したか否かを判定する手段であるものである。

## 【0015】

この発明に係る情報処理装置は、検出手段が、リピートブロックを繰り返し処理する回数をカウントするカウント手段と、上記リピートブロック内の実行される命令のアドレスと最後に実行される最終命令のアドレスとを比較する比較手段とを有しており、上記カウント手段のカウント数が所定の数に到達し、上記比較手段が実行される命令のアドレスが最終命令のアドレスと一致したことを通知した際に繰り返し処理の終了と判定するものである。

## 【0016】

この発明に係る情報処理装置は、検出手段が、リピートブロックの繰り返し処理時に実行される命令数に基づき繰り返し処理が終了したか否かを判定する手段であるものである。

## 【0017】

この発明に係る情報処理装置は、検出手段が、リピートブロック内の実行された命令数をカウントするカウント手段を有しており、上記カウント手段のカウント数が所定の数に到達した際に繰り返し処理の終了と判定するものである。

## 【0018】

この発明に係る情報処理装置は、検出手段が、リピートブロックを繰り返し処理する回数をカウントする第1カウント手段と、上記リピートブロック内の実行された命令数をカウントする第2カウント手段とを有しており、上記第1カウント手段のカウント数が第1の所定の数に到達し、上記リピートブロックの最終の繰り返し処理において上記第2カウント手段のカウント数が第2の所定の数に到達した際に繰り返し処理の終了と判定するものである。

## 【0019】

## 【発明の実施の形態】

以下、この発明の実施の一形態を説明する。

実施の形態 1.

図 1 はこの発明の実施の形態 1 による情報処理装置のレジスタセットを示す図である。なお、以下では、この発明による情報処理装置は、16ビットプロセッサであり、アドレスおよびデータのビット長は16ビットであると仮定する。また、この発明による情報処理装置はビット順およびバイト順に関してビッグエンディアンを採用しており、ビット位置についてMSBがビット0である。

#### 【0020】

図 1 において、1～15はそれぞれデータまたはアドレス値を保持する汎用レジスタR0～R14である。なお、汎用レジスタR13は、サブルーチンジャンプ時の戻り先アドレスを保持するためのリンク（LINK）レジスタである。また、汎用レジスタR15はスタックポインタ（SP）を保持するためのものであり、割り込み用のスタックポインタSPI16とユーザ用のスタックポインタSPU17とから成る。汎用レジスタR15は、後で説明するプロセッサ・ステータス・ワード（PSW）によって切り替えられてSPI16またはSPU17を選択する。以後、SPI16およびSPU17を総称して、スタックポインタ（SP）と呼ぶ。特別な場合を除き、4ビットのレジスタ指定フィールドでオペランドとなる汎用レジスタの番号が指定される。この発明による情報処理装置では、例えば汎用レジスタR0 1, R1 2のように2つの汎用レジスタをペアにして指定する命令が設けられている。この場合、単に偶数番号の汎用レジスタを指定することにより、これとペアの汎用レジスタとして、そのレジスタ番号を1プラスした奇数番号の汎用レジスタが暗に指定される。

#### 【0021】

21～30はそれぞれ16ビットの制御レジスタCR0～CR3, CR6～CR11である。各制御レジスタの番号も汎用レジスタと同様に4ビットのレジスタ指定フィールドで指定される。制御レジスタCR0 21は、情報処理装置の動作モードを指定するビットや演算結果を示すフラグからなるプロセッサ・ステータス・ワード（PSW）を保持するレジスタである。

#### 【0022】

図 2 は制御レジスタCR0 21に保持されるPSWの構成を示す図である。

図において、41すなわちビット0はスタックモードを示すSMビットである。SMビット41が“0”の場合は割り込みモードであることを示しており、汎用レジスタR15としてSPI16が選択される。他方、SMビット41が“1”の場合はユーザモードであることを示しており、汎用レジスタR15としてSPU17が選択される。42すなわちビット5は割り込みイネーブルを指定するIEビットであり、IEビット42が“0”の場合は割り込みはマスク（アサートされても無視）され、IEビット42が“1”の場合は割り込みは受け付けられる。

### 【0023】

また、この情報処理装置は、ゼロオーバーヘッドのリピート処理を実現するためのリピート機能を備えている。図2において、43すなわちビット6はリピート状態を示すRPビットであり、RPビット43が“0”の場合はリピート中ではないことを示しており、RPビット43が“1”の場合はリピート中であることを示している。この情報処理装置では、さらに、サーキュラーバッファをアクセスするためのアドレッシングであるモジュロアドレッシング機能がインプリメントされている。44すなわちビット7はモジュロイネーブルを指定するMDビットであり、MDビット44が“0”の場合はモジュロアドレッシングはディスエーブル状態にされ、MDビット44が“1”の場合はモジュロアドレッシングはイネーブル状態にされる。また、45すなわちビット8は、アキュムレータのデータフォーマットを指定するFXビットであり、FXビット45が“0”の場合乗算結果は整数フォーマットでアキュムレータに格納され、FXビット45が“1”の場合乗算結果は固定小数点フォーマットとして1ビット左にシフトされてアキュムレータに格納される。46すなわちビット9はサチュレーションモードを指定するSTビットであり、STビット46が“0”の場合アキュムレータに演算結果を格納する際、演算結果が40ビットとして書き込まれ、STビット46が“1”の場合アキュムレータに演算結果を格納する際、演算結果は32ビットで表現できる値にリミット処理されて書き込まれる。例えば、演算結果が16進で007fffffより大きい値の場合には、アキュムレータには007fffffが書き込まれ、演算結果が16進でff800000より小



さい値の場合には、アキュムレータには f f 8 0 0 0 0 0 0 0 が書き込まれる。

#### 【0024】

また、47すなわちビット12は実行制御フラグ（F0フラグ）であり、比較命令の比較結果などがこのフラグにセットされる。48すなわちビット13も実行制御フラグ（F1フラグ）であり、比較命令等によりF0フラグ47が更新される際に、更新前のF0フラグ47の値が、F1フラグにコピーされる。また、49すなわちビット15はキャリーフラグであり、加減算命令実行時のキャリーがこのキャリーフラグにセットされる。

#### 【0025】

図1において、制御レジスタCR2 23はプログラムカウンタ（PC）であり、実行中の命令アドレスを示すものである。この発明による情報処理装置が処理する命令は、基本的に32ビット固定長であり、制御レジスタCR2 23は32ビットを1ワードとした命令ワードアドレスを保持する。制御レジスタCR1 22はバックアップ・プロセッサ・ステータス・ワード（BPSW）を保持するレジスタであり、制御レジスタCR3 24はバックアップ・プログラム・カウンタ（BPC）であり、これらはそれぞれ例外や割り込みが検出された場合の実行中のPSWの値とPCの値を待避、保持するためのレジスタである。また、制御レジスタCR6～CR9 25～28はリピート関連のレジスタであり、リピート中であっても割り込みを受け付けられるように、ユーザに値の読み書きを可能にするためのものである。制御レジスタCR7 26は、繰り返し回数を示すカウント値RPT\_Cを保持するリピート・カウンタ（RPT\_C）レジスタであり、制御レジスタCR8 27は、リピートブロックの先頭の命令（すなわち開始命令）のアドレスRPT\_Sを保持するリピート・ブロック・スタート・アドレス（RPT\_S）レジスタであり、制御レジスタCR9 28は、リピートブロックの最後の命令（以下リピートブロック最終命令と称する）のアドレスRPT\_Eを保持するリピート・ブロック・エンド・アドレス（RPT\_E）レジスタであり、制御レジスタCR6 25は、リピートブロック内でリピート処理を終了する最終命令（以下リピート処理最終命令と称する）のアドレスRPT\_Bを保持するリピート・ブレーク・アドレス（RPT\_B）レジスタである。

## 【0026】

また、制御レジスタCR10 29および制御レジスタCR11 30は、モジュロ・アドレッシングを行うために使用されるレジスタであり、制御レジスタCR10 29はモジュロ・スタート・アドレス(MOD\_\_S)を保持し、制御レジスタCR11 30はモジュロ・エンド・アドレス(MOD\_\_E)を保持する。これらの制御レジスタは共にデータ・ワード(16ビット)・アドレスを保持する。ポストインクリメントでアドレスを更新するモジュロアドレッシングを利用する場合には、制御レジスタCR10 29に小さい方のアドレスがセットされ、制御レジスタCR11 30に大きい方のアドレスがセットされる。そして、インクリメント前のレジスタ値が制御レジスタCR11 30に保持されているアドレスと一致した場合、インクリメント結果として制御レジスタCR10 29に保持されている値が汎用レジスタに書き戻される。

## 【0027】

図1において、31, 32はそれぞれ40ビットのアキュムレータA0, A1である。アキュムレータA0 31は、積和演算結果の上位16ビットを保持するA0H 31b、積和演算結果の下位16ビットを保持するA0L 31c、積和演算結果の上位からあふれたビットを保持する8ビットのガードビットA0G 31aからなる。同様に、アキュムレータA1 32は、積和演算結果の上位16ビットを保持するA1H 32b、積和演算結果の下位16ビットを保持するA1L 32c、積和演算結果の上位からあふれたビットを保持する8ビットのガードビットA1G 32aからなる。

## 【0028】

この発明による情報処理装置は2ウェイのVLIW (Very Long Instruction Word) 命令セットを処理する。図3はこの発明による情報処理装置の32ビット命令のフォーマットを示す図である。基本命令長は32ビット固定であり、各命令は32ビット単位で整置されている。32ビットの各命令コードは、命令のフォーマットを示す2ビットのフォーマット指定ビット(FMビット)51と、15ビットの左コンテナ52および15ビットの右コ

ンテナ 53 とから構成される。各コンテナ 52, 53 はそれぞれ 15 ビットのショートフォーマットのサブ命令を格納できるほか、2 つで 1 つの 30 ビットのロングフォーマットのサブ命令を格納できる。なお、以下では、ショートフォーマットのサブ命令をショート命令、ロングフォーマットのサブ命令をロング命令と呼ぶ。

## 【0029】

FM ビット 51 は命令のフォーマットおよび 2 つのショート命令の実行順序を指定する。FM ビット 51 が “11” の場合は、左コンテナ 52 および右コンテナ 53 の 30 ビットで 1 つのロング命令を保持していることを示しており、それ以外の場合は左コンテナ 52 および右コンテナ 53 がそれぞれショート命令を保持していることを示している。さらに、命令コードが 2 つのショート命令を保持している場合、その FM ビット 51 が実行順序を指定している。FM ビット 51 が “00” の場合、2 つのショート命令を並列に実行することを示している。FM ビット 51 が “01” の場合、左コンテナ 52 に保持されているショート命令を実行した後に、右コンテナ 53 に保持されているショート命令を実行することを示している。また、FM ビット 51 が “10” の場合、右コンテナ 53 に保持されているショート命令を実行した後に、左コンテナ 52 に保持されているショート命令を実行することを示している。このように、この発明による情報処理装置は、シーケンシャルに実行する 2 つのショート命令も含めて 1 つの 32 ビット命令にエンコードしてコード効率の向上を図っている。

## 【0030】

図 4 ～ 図 7 は典型的な命令のビット割り付けの例を示す図である。図 4 は 2 つのオペランドを持つショート命令のビット割り付けの例を示している。図において、61, 64 はオペレーションコードフィールドである。これに代わって、フィールド 64 はアキュムレータ番号を指定してもよい。また、62, 63 は、オペランドとして参照あるいは更新されるデータの格納位置をレジスタ番号やアキュムレータ番号で指定するフィールドである。これに代わって、フィールド 63 は 4 ビットの小さな即値データを指定してもよい。図 5 はショートフォーマットの分岐命令のビット割り付けの例を示しており、図において、71 はオペレーシ

ンコードフィールドであり、72は8ビットの分岐変位フィールドである。分岐変位は、PC値と同様、命令ワード(32ビット)のオフセットで指定される。図6は16ビットの変位や即値データを持つ3オペランド命令やロード/ストア命令のビット割り付けの例を示しており、図において、81はオペレーションコードフィールドであり、82, 83はショートフォーマットと同様にレジスタ番号を指定するフィールドであり、84は変位や即値データ等を指定する16ビットの拡張データフィールドである。図7は右コンテナ53側にオペレーションコードを持つロングフォーマットの命令のビット割り付けの例を示しており、2ビットのフィールド91が“01”である例である。93, 96はオペレーションコードフィールドであり、94, 95はレジスタ番号等を指定するフィールドである。また、92は予約フィールドであり、必要に応じてオペレーションコードやレジスタ番号等の指定に使用される。

#### 【0031】

上述以外に、NOP(ノーオペレーション)命令のように、15ビットすべてがオペレーションコードとなる命令や、1オペランド命令等の特殊なビット割り付けを持つ命令がある。

#### 【0032】

この発明による情報処理装置の各サブ命令はRISCライクな命令セットである。メモリデータのアクセスを行う命令はロード/ストア命令のみであり、演算命令はレジスタ/アキュムレータ中のオペランドや即値オペランドに対して演算を行う。オペランドデータのアドレッシングモードとしては、レジスタ間接モード、ポストインクリメント付きレジスタ間接モード、ポストデクリメント付きレジスタ間接モード、プッシュモード、レジスタ相対間接モードの5種類がある。各々のニーモニックは、“@Rsrc”、“@Rsrc+”、“@Rsrc-”、“@-SP”、“@ (disp16, Rsrc)”で示される。Rsrcはベースアドレスを指定するレジスタ番号を示しており、disp16は16ビットの変位値を示している。また、オペランドのアドレスはバイトアドレスで指定される。

#### 【0033】

レジスタ相対間接モード以外のロード／ストア命令は、図4に示す命令フォーマットを有する。かかるロード／ストア命令では、フィールド63でベースレジスタ番号が指定され、フィールド62でメモリからロードしてきた値を書き込むレジスタの番号もしくはストアする値を保持するレジスタの番号が指定される。レジスタ間接モードでは、ベースレジスタとして指定されたレジスタの値がオペランドアドレスとなる。ポストインクリメント付きレジスタ間接モードでは、ベースレジスタとして指定されたレジスタの値がオペランドアドレスとなり、このベースレジスタの値がオペランドのサイズ（バイト数）分ポストインクリメントされて書き戻される。ポストデクリメント付きレジスタ間接モードでは、ベースレジスタとして指定されたレジスタの値がオペランドアドレスとなり、このベースレジスタの値がオペランドのサイズ（バイト数）分ポストデクリメントされて書き戻される。プッシュモードは、ストア命令の実行で且つベースレジスタが汎用レジスタR15である場合にのみ使用可能であり、スタックポインタ（SP）値がオペランドのサイズ（バイト数）分プリデクリメントされた値が、オペランドアドレスとなり、デクリメントされた値がSPに書き戻される。

#### 【0034】

レジスタ相対間接モードのロード／ストア命令は図6に示す命令フォーマットを有している。かかるロード／ストア命令では、フィールド83でベースレジスタ番号が指定され、フィールド82でメモリからロードしてきた値を書き込むレジスタの番号もしくはストアする値を保持するレジスタの番号が指定される。また、フィールド84はオペランド格納位置のベースアドレスからの変位値を指定する。レジスタ相対間接モードでは、ベースレジスタとして指定されたレジスタの値に16ビットの変位値を加算した値がオペランドアドレスとなる。

#### 【0035】

ポストインクリメント付きレジスタ間接モードおよびポストデクリメント付きレジスタ間接モードでは、制御レジスタCR0 21に保持されたPSW中のMDビット44を“1”にすることにより、モジュロアドレッシングモードが使用され得る。

#### 【0036】

ジャンプ命令のジャンプ先アドレス指定には、ジャンプ先アドレスをレジスタ値で指定するレジスタ間接モードと、ジャンプ命令のPCからの分岐変位で指定するPC相対間接モードとがある。さらに、PC相対間接モードについては、分岐変位を8ビットで指定するショートフォーマットと、分岐変位を16ビットで指定するロングフォーマットの2種類ある。また、オーバーヘッドなしにリピート処理を実現するリピート機能を起動するためのリピート命令も設けられている。

#### 【0037】

図8はこの発明の実施の形態1による情報処理装置100の構成を示すブロック図である。図において、101はMPUコア部であり、102はMPUコア部101からの要求により命令データのアクセスを行う命令フェッチ部であり、103は命令データを格納するための内蔵命令メモリであり、104はMPUコア部101からの要求によりオペランドデータのアクセスを行うオペランドアクセス部であり、105はオペランドデータを格納するための内蔵データメモリであり、106は命令フェッチ部102およびオペランドアクセス部104からの要求を調停し、情報処理装置100の外部のメモリへアクセスする外部バスインタフェース部である。

#### 【0038】

また、MPUコア部101は、命令キュー111と、制御部（検出手段、命令実行抑止手段、命令処理シーケンス切替手段）112と、レジスタファイル115と、第1演算部116と、第2演算部117と、PC部（検出手段、命令実行抑止手段、命令処理シーケンス切替手段）118とを備えている。

#### 【0039】

命令キュー111は、2エントリ分の32ビット命令バッファおよび有効ビットと、入出力ポインタとを有しており、FIFO（先入れ先出し）方式で制御される。命令キュー111は、命令フェッチ部102でフェッチされた命令データを一時保持し、制御部112に送る。

#### 【0040】

制御部112は、命令キュー111の制御、パイプライン制御、命令の実行制

御、命令フェッチ部102やオペランドアクセス部104とのインターフェイス制御など、MPUコア部101に含まれるすべての構成要素の制御を行う。また、制御部112は命令キュー111から送られる命令コードをデコードするための命令デコード部119を備えており、命令デコード部119は第1デコーダ113および第2デコーダ114を含む。第1デコーダ113は第1演算部116により実行される命令をデコードし、第2デコーダ114は第2演算部117により実行される命令をデコードする。32ビットの命令のデコードの第1サイクルでは、必ず左コンテナ52の命令コードが第1デコーダ113により解析され、右コンテナ53の命令コードが第2デコーダ114により解析される。ただし、FMビット51および左コンテナのビット0とビット1のデータは両方のデコーダ113、114により解析される。また、右コンテナ53のデータから拡張データを抽出するために、右コンテナ53のデータは第1デコーダ113に送られるが、解析は行われぬ。従って、最初に実行する命令はその命令を実行する演算部に対応した位置すなわち左コンテナ52または右コンテナ53に置かれなければならない。2つのショート命令をシーケンシャルに実行する場合、先行して実行される命令のデコード中に後で実行される命令が図示していないプリデコーダでデコードされ、そのプリデコーダによりどちらのデコーダでデコードすべきかが判定される。後で実行される命令がどちらのデコーダでも処理できる命令の場合は、第1デコーダ113によりデコードされる。先行する命令のデコード後、後で実行する命令の命令コードが選択されたデコーダに取り込まれ、解析される。

#### 【0041】

レジスタファイル115は、汎用レジスタR0～R15に相当する複数のレジスタを有しており、第1演算部116、第2演算部117、PC部118およびオペランドアクセス部104に複数のバスを介して接続されている。

#### 【0042】

図9は第1演算部116の構成を詳細に示すブロック図である。図に示すように、第1演算部116は、S1バス301、S2バス302およびS3バス303を介してレジスタファイル115に接続されている。第1演算部116は、こ

れら3つのバスを介してレジスタファイル115からデータを読み出し、内部の演算器等にリードオペランドとなるデータやストアデータを転送する。S1バス301はレジスタファイル115内の偶数番号のレジスタのみに接続されており、S2バス302は奇数番号のレジスタのみに接続されており、第1演算部116はS1バス301およびS2バス302を介してレジスタペアの2ワードを並列にレジスタファイル115から読み出すこともできる。S3バス303はレジスタファイル115内の全レジスタに接続されている。

## 【0043】

第1演算部116は、また、D1バス311およびWバス314を介してレジスタファイル115に接続されており、第1演算部116は、D1バス311を介して演算結果やデータをレジスタファイル115へ転送し、Wバス314を介してロードしたバイトデータをレジスタファイル115へ転送する。D1バス311およびWバス314は、レジスタファイル115内の全レジスタに接続されている。

## 【0044】

レジスタファイル115は32ビットのODバス322を介してオペランドアクセス部104に接続されており、情報処理装置は1ワードのデータまたはレジスタペアの2ワードを並列にオペランドアクセス部104からレジスタファイル115へ転送することが可能である。ODバス322の上位16ビットおよび下位16ビットは、ともにレジスタファイル115の任意のレジスタにデータ等を書き込むことができるようにレジスタファイル115内の全レジスタに接続されている。

## 【0045】

また、図9において、153は主として転送、比較、算術論理演算、オペランドアドレスの計算／転送、オペランドアドレス値のインクリメント／デクリメント、ジャンプ先アドレスの計算／転送等を行う算術論理演算部（ALU）であり、151、152はALU153の入力ラッチであるAAラッチ、ABラッチである。AAラッチ151は、S1バス301、S2バス302またはS3バス303を介して読み出されたレジスタ値を取り込む。AAラッチ151は保持する



内容をゼロクリアする機能も備えている。ABラッチ152は、S3バス303を介して読み出されたレジスタ値もしくは第1デコーダ113によるデコードの結果生成された16ビットの即値データを取り込む。ABラッチ152は保持する内容をゼロクリアする機能も備えている。

## 【0046】

上記したように、ALU153は、主として転送、比較、算術論理演算、オペランドアドレスの計算／転送、オペランドアドレス値のインクリメント／デクリメント、ジャンプ先アドレスの計算／転送等を行う。ALU153は、また、演算結果やアドレスモディファイの結果をセクタ155およびD1バス311を介してレジスタファイル115中の実行される命令で指定されたレジスタに書き戻す。

## 【0047】

また、154はオペランドのアドレスを保持するAOラッチであり、AOラッチ154はALU153によるアドレス計算結果またはAAラッチ151に保持されたベースアドレスの値を選択的に保持し、保持した値をOAバス321を介してオペランドアクセス部104に出力する。また、ALU153がジャンプ先アドレスやリピートエンドアドレスなどを計算した場合には、ALU153は出力をJAバス323を介してPC部118へ転送する。

## 【0048】

さらに、156、157はそれぞれ図1に示す制御レジスタCR10 29および制御レジスタCR11 30に相当するMOD\_Sレジスタ、MOD\_Eレジスタであり、158はMOD\_Eレジスタ157に保持されている値とS3バス303上のベースアドレスの値とを比較する比較器(CMP)である。MOD\_Sレジスタ156はラッチ159を介してセクタ155に接続されている。

## 【0049】

また、169はセクタ155と協働して、S1バス301およびS2バス302とD1バス311との間に転送経路を形成するラッチである。

## 【0050】

さらに、160は2つの16ビットのレジスタからなり、S1バス301もし

くはS2バス302に、または、S1バス301およびS2バス302の両方に出力されたストアデータを一時保持するストアデータ(SD)レジスタであり、162はラッチ161を介して転送されたSDレジスタ160に保持されていたストアデータをオペランドのアドレスに従って32ビット境界に整置する整置回路である。整置後のストアデータは、ラッチ163およびODバス322を介してオペランドアクセス部104に出力される。

#### 【0051】

また、164はオペランドアクセス部104によりロードされたバイトデータをODバス322を介して取り込む16ビットのロードデータ(LD)レジスタである。LDレジスタ164が取り込んだデータはラッチ165を介して整置回路166へ転送される。整置回路166は、そのデータに対してバイト整置を行い、さらにバイトデータのゼロ/符号拡張を行う。そして、整置、拡張後のデータは、Wバス314を介してレジスタファイル115中の指定されたレジスタに書き込まれる。1ワード(16ビット)ロードまたは2ワード(32ビット)ロードの場合には、LDレジスタ164を介さず、ロードした値がODバス322からレジスタファイル115へ直接書き込まれる。

#### 【0052】

221は制御部112に含まれ、図1に示す制御レジスタCR0 21の内容を保持するPSWラッチ222等のラッチやPSW更新回路等からなり、演算結果や命令の実行に応じてPSWラッチ222に保持されている値を更新するPSW部である。情報処理装置は、PSWラッチ222に値を転送する場合、TPSWラッチ167を介して、S3バス303に出力されたデータの必要なビット(アサインされているビット)のみを転送する。情報処理装置は、PSWラッチ222に保持されている値を読み出す場合には、PSW部221からその値をD1バス311に出力し、レジスタファイル115へ書き込む。また、168は図1に示す制御レジスタCR1 22に相当するBPSWレジスタである。例外処理時には、D1バス311に出力されたPSWラッチ222に保持されている値がBPSWレジスタ168に書き込まれる。BPSWレジスタ168に書き込まれた値をレジスタファイル115へ転送する等の場合には、BPSWレジスタ16

8に保持されている値はS3バス303上に読み出された後レジスタファイル115等の必要な場所へ転送される。この際、アサインされていないビットは強制的にゼロがS3バス303に出力される。例外からの復帰時には、BPSWレジスタ168に保持されている値は、直接TPSWラッチ167を介して、必要なビット（アサインされているビット）のみPSW部221のPSWラッチ222へ転送される。

#### 【0053】

図10はPC部118の構成を詳細に示すブロック図である。図において、181は次にフェッチする命令のアドレスすなわち命令フェッチアドレスを保持し、そのアドレスを命令フェッチ部102に出力する命令アドレス（IA）レジスタであり、183は命令フェッチ部102が引き続き後続の命令をフェッチすることを可能にするために、IAレジスタ181からラッチ182を介して転送されたアドレス値を1インクリメントするインクリメンタ（INC）である。インクリメンタ183でインクリメントされたアドレス値はIAレジスタ181に書き戻される。ジャンプやリピート等によりシーケンスが切り替わる場合には、IAレジスタ181は、JAバス323を介して転送されるジャンプ先アドレス、リピートブロック開始アドレス、リピートブロックの次の命令のアドレスを取り込む。

#### 【0054】

また、184、186、188、171は、それぞれリピート制御用の制御レジスタであるRPT\_\_Sレジスタ、RPT\_\_Eレジスタ、RPT\_\_Cレジスタ、RPT\_\_Bレジスタであり、それぞれ図1に示す制御レジスタCR8、CR9、CR7、CR6に相当しており、それぞれRPT\_\_S、RPT\_\_E、RPT\_\_C、RPT\_\_Bを保持する。RPT\_\_Sレジスタ184、RPT\_\_Eレジスタ186、RPT\_\_Cレジスタ188、RPT\_\_Bレジスタ171は、それぞれD1バス311に接続された入力ポートとS3バス303に接続された出力ポートを有しており、各レジスタには必要に応じてリピート時の初期設定や待避復帰が行なわれる。

#### 【0055】

RPT\_\_Sレジスタ184はリピートブロックの開始命令のアドレスRPT\_\_Sを保持する。RPT\_\_Sレジスタ184の初期設定直後には、ラッチ185の内容も更新される。リピート処理中にリピートブロックの先頭命令に戻る場合は、ラッチ185に保持されている値はJAバス323を介してIAレジスタ181へ転送される。

## 【0056】

RPT\_\_Eレジスタ186はリピートブロック最終命令のアドレスRPT\_\_Eを保持する。このリピートブロック最終命令のアドレスは、リピート命令処理時に第1演算部116により計算され、JAバス323を介してRPT\_\_Eレジスタ186に取り込まれる。

## 【0057】

187はRPT\_\_Eレジスタ186に保持されている値と、IAレジスタ181に保持されている命令フェッチアドレスとを比較し、これらの値が一致しているか否かを示す一致情報を出力する比較器（CMP）であり、173はRPT\_\_Eレジスタ186の内容の更新時に、RPT\_\_Eレジスタ186に保持されている値を1インクリメントしてリピートブロックの次の命令のアドレスを計算するインクリメンタ（INC）であり、インクリメントされた値はラッチ174に書き込まれる。リピート処理完了時には、ラッチ174に保持されている値はJAバス323を介してIAレジスタ181へ転送される。

## 【0058】

RPT\_\_Bレジスタ171はリピート処理を終了（ブレイク）するリピートブレイクアドレスRPT\_\_Bを保持する。このリピートブレイクアドレスは、リピート命令処理時に第1演算部116により計算され、JAバス323を介してRPT\_\_Bレジスタ171に取り込まれる。172はRPT\_\_Bレジスタ171に保持されている値と、IAレジスタ181に保持されている命令フェッチアドレスとを比較し、これらの値が一致しているか否かを示す一致情報を出力する比較器（CMP）である。この比較器172は比較手段に相当する。

## 【0059】

RPT\_\_Cレジスタ188およびTRPT\_\_Cレジスタ175は、それぞれリ

ピートブロックの繰り返し回数を管理するためのカウント値RPT\_Cを保持する。TRPT\_Cレジスタ175は、D1バス311に接続された入力ポートを備えており、RPT\_Cレジスタ188の初期設定と同時に初期化される。なお、TRPT\_Cレジスタ175に保持されているカウント値RPT\_Cを先行更新情報と呼ぶ。

#### 【0060】

また、177は命令フェッチ段階で繰り返し回数を更新する場合、ラッチ176を介して転送されてきたTRPT\_Cレジスタ175に保持されている値を1デクリメントするデクリメンタ(DEC)であり、1デクリメントされた値はTRPT\_Cレジスタ175に書き戻される。また、178はTRPT\_Cレジスタ175に保持されている値がゼロであるか否かを検出し、検出結果を出力するゼロ検出回路(ZERO)である。TRPT\_Cレジスタ175に保持されている値は、ラッチ176を介してRPT\_Cレジスタ188へ転送される。また、179はジャンプが起こった場合にTRPT\_Cレジスタ175に保持されている値を初期化するために、RPT\_Cレジスタ188からTRPT\_Cレジスタ175へ至る転送経路を形成するラッチである。なお、カウント手段は、TRPT\_Cレジスタ175、ラッチ176およびデクリメンタ177からなる。

#### 【0061】

さらに、194は実行中の命令のPC値を保持する実行ステージプログラムカウンタ(EPC)であり、191は次に実行する命令のPC値を保持する次命令プログラムカウンタ(NPC)である。NPC191は、実行段階でジャンプが起こった場合、JAバス323上のジャンプ先アドレス値を取り込む。リピートブロックの処理を繰り返す場合には、NPC191はラッチ185からリピートブロックの先頭アドレスを取り込む。他方、リピート処理を終了する場合には、NPC191はラッチ174からリピートブロックの次の命令のアドレスを取り込む。それ以外の場合には、ラッチ192を介して転送されたNPC191に保持されている値はインクリメンタ(INC)193により1インクリメントされNPC191に書き戻される。サブルーチンジャンプ命令の場合には、ラッチ192に保持されている値が戻り先アドレスとしてD1バス311に出力され、レ

レジスタファイル 1 1 5 中のリンクレジスタとして定義されている汎用レジスタ R 1 3 1 4 に書き込まれる。次に実行する命令の P C 値を参照する場合には、N P C 1 9 1 に保持されている値が S 3 バス 3 0 3 に出力され、第 1 演算部 1 1 6 へ転送される。また、次の命令が実行状態に入る場合には、ラッチ 1 9 2 に保持されている値が E P C 1 9 4 へ転送される。情報処理装置は、実行中の命令の P C 値を参照する場合、E P C 1 9 4 に保持されている値を S 3 バス 3 0 3 に出力し、第 1 演算部 1 1 6 へ転送する。

#### 【 0 0 6 2 】

1 9 6 は図 1 に示すレジスタセットに含まれる制御レジスタ C R 3 2 4 に対応するバックアッププログラムカウンタ ( B P C ) である。例外や割り込み等が検出された場合には、E P C 1 9 4 に保持されている値がラッチ 1 9 5 を介して B P C 1 9 6 へ転送される。B P C 1 9 6 は D 1 バス 3 1 1 に接続された入力ポートと S 3 バス 3 0 3 に接続された出力ポートとを有しており、B P C 1 9 6 は必要に応じて待避復帰される。

#### 【 0 0 6 3 】

図 1 1 は第 2 演算部 1 1 7 の構成を詳細に示すブロック図である。図 1 1 に示すように、第 2 演算部 1 1 7 は、S 4 バス 3 0 4 および S 5 バス 3 0 5 を介してレジスタファイル 1 1 5 に接続されており、これら 2 つのバスを介して 2 つの任意のレジスタからデータを読み出すことができる。第 2 演算部 1 1 7 は、また、S 4 バス 3 0 4 および S 5 バス 3 0 5 を介してレジスタファイル 1 1 5 中の任意のレジスタペアの 2 ワードを並列に読み出すことも可能である。第 2 演算部 1 1 7 は、さらに、D 2 バス 3 1 2 および D 3 バス 3 1 3 を介してレジスタファイル 1 1 5 に接続されており、これら 2 つのバスのいずれかを介してデータをレジスタファイル 1 1 5 中の任意のレジスタに書き込むことができる。D 2 バス 3 1 2 はレジスタファイル 1 1 5 中の偶数番号のレジスタのみに接続されており、D 3 バス 3 1 3 はレジスタファイル 1 1 5 中の奇数番号のレジスタのみに接続されている。第 2 演算部 1 1 7 は、また、D 2 バス 3 1 2 および D 3 バス 3 1 3 を介してレジスタファイル 1 1 5 中の任意のレジスタペアに 2 ワードのデータを並列に書き込むこともできる。

## 【0064】

図11において、208は図1に示す2つの40ビットアキュムレータA0、A1 31、32に相当するアキュムレータであり、201は40ビットのALUである。ALU201は、ビット0からビット7までの8ビットがアキュムレータ208のガードビット用加算器、ビット8からビット23までの16ビットが算術論理演算器、ビット24からビット39までの16ビットがアキュムレータ208の下位16ビットを加算するための加算器になっており、40ビットまでの加減算と16ビットの論理演算とを行うことができる。

## 【0065】

また、202、203はともにALU201の40ビット入力ラッチであるAラッチ、Bラッチであり、204はアキュムレータ208に保持されている値をそのままもしくは16ビット算術右シフトして出力するシフタである。Aラッチ202は、S4バス304からレジスタ値をALU201の入力のビット8からビット23の位置に取り込むか、シフタ204を介して、アキュムレータ208に保持されている値をそのままもしくは16ビット算術右シフトした値を取り込む。

## 【0066】

さらに、205はアキュムレータ208に保持されている値を配線206（ガードビット8ビット）、S4バス304（上位16ビット）およびS5バス305（下位16ビット）を介して取り込むか、レジスタファイル115中の任意のレジスタに保持されている値をS5バス305のみ、もしくは、S4バス304およびS5バス305の両方を介して16ビットもしくは32ビットのデータを右詰めで取り込み、40ビットに符号拡張し、入力データを左3ビットから右2ビットの任意のシフト量で算術シフトして出力するシフタである。Bラッチ203は、S5バス305上のデータをALU201の入力のビット8からビット23の位置に取り込むか、または、後述する乗算器211のPラッチ214の値もしくはシフタ205の出力を取り込む。なお、Aラッチ202およびBラッチ203は、それぞれ保持する値をゼロクリアしたり定数の値に設定したりする機能を有する。

## 【0067】

209はALU201の40ビットの出力を受け、この出力を上位16ビットもしくは上位下位あわせた32ビットのデータに変換する際に、ガードビットをチェックして、ALU201の出力の上位20ビットまたは40ビットの出力全体を16ビットまたは32ビットで表現できる最大値もしくは最小値にクリッピングする機能および入力をそのまま出力する機能を備えたサチュレーション回路であり、207はマルチプレクサ(MUX)である。サチュレーション回路209の出力はマルチプレクサ207に接続されている。

## 【0068】

デスティネーションオペランドがアキュムレータ208を示す場合には、マルチプレクサ207の出力はアキュムレータ208に書き込まれる。デスティネーションオペランドがレジスタを示す場合は、マルチプレクサ207の出力はD2バス312やD3バス313を介してレジスタファイル115の指定したレジスタに書き込まれる。1ワード転送の場合にはデスティネーションレジスタの番号が偶数ならば、その1ワードデータはD2バス312に出力され、奇数ならばD3バス313に出力される。2ワード転送の場合には、上位側の16ビットデータがD2バス312に出力され、下位側の16ビットデータがD3バス313に出力される。また、転送命令、絶対値の計算命令、最大値設定命令または最小値設定命令を実行するために、Aラッチ202およびBラッチ203の出力はマルチプレクサ207に接続されており、Aラッチ202およびBラッチ203に保持されている値をアキュムレータ208やレジスタファイル115へ転送することが可能である。

## 【0069】

また、210はBラッチ203に保持されている値を取り込み、固定小数点フォーマットの数を正規化するのに必要なシフト量を計算し、レジスタファイル115に書き戻すために結果をD2バス312またはD3バス313に出力するプライオリティエンコーダである。さらに、212、213はそれぞれ乗算器211の入力レジスタとして働き、S4バス304およびS5バス305の16ビットの値を取り込み、17ビットにゼロ拡張または符号拡張する機能を備えるXラ



ッチ、Yラッチである。乗算器211は、17ビット×17ビットの乗算器であり、Xラッチ212に保持されている値とYラッチ213に保持されている値との乗算を行う。積和命令や積差命令の場合には、第2演算部117は乗算器211による乗算結果をPラッチ214に書き込みBラッチ203へ転送する。乗算命令でデスティネーションオペランドがアキュムレータ208を示す場合には、第2演算部117は乗算器211による乗算結果をマルチプレクサ207を介してアキュムレータ208に書き込む。

#### 【0070】

215は40ビットまたは16ビットのデータに対して、左右16ビットまでの算術／論理シフトを行うバレルシフタであり、216は即値データまたはレジスタ値をS5バス305を介してシフト量として取り込むシフト量(SC)ラッチであり、217は、アキュムレータ208に保持されている値またはS4バス304上に出力されたレジスタに保持されている値をシフトデータとして取り込むシフトデータ(SD)ラッチである。バレルシフタ215は、SDラッチ217に保持されたシフトデータをSCラッチ216で指定されるシフト量だけ、オペレーションコードで指定されたシフトを行う。シフト結果はサチュレーション回路209に出力され、ALU201による演算結果と同様、必要に応じてサチュレーションが行われた後マルチプレクサ207に出力される。マルチプレクサ207に出力された値は、アキュムレータ208に書き戻されるか、または、D2バス312およびD3バス313を介してレジスタファイル115に書き戻される。

#### 【0071】

また、218は第2デコーダ114により生成された6ビットの即値データを16ビットに拡張して保持し、S5バス305を介して第2演算部117へ転送するための即値ラッチである。即値ラッチ218は、また、ビット操作命令のビットマスクも生成する。

#### 【0072】

次に動作について説明する。

まず、この発明の実施の形態1による情報処理装置のパイプライン処理につい

て説明する。図12は実施の形態1による情報処理装置のパイプライン処理を示す図である。情報処理装置は、命令データのフェッチを行う命令フェッチ（IF）ステージ401、命令の解析を行う命令デコード（D）ステージ402、演算実行を行う命令実行（E）ステージ403、データメモリのアクセスを行うメモリアクセス（M）ステージ404、メモリからロードしたバイトオペランドをレジスタへ書き込むライトバック（W）ステージ405の5段のパイプライン処理を行う。Eステージ403での演算結果のレジスタへの書き込みはEステージ403で完了する。また、ワード（2バイト）、2ワード（4バイト）ロード時のレジスタへの書き込みはMステージ404で完了する。積和／積差演算に関する命令は、更に乗算と加算の2段のパイプラインで実行される。後段の処理を命令実行2（E2）ステージ406と呼ぶ。連続する積和／積差演算を1回／1クロックサイクルのスループットで実行できる。

#### 【0073】

IFステージ401では、主して命令のフェッチ、命令キュー111の管理、リピート制御が行われる。命令フェッチ部102、内蔵命令メモリ103、外部バスインタフェース部106、命令キュー111、PC部118の一部（IAレジスタ181、ラッチ182、インクリメンタ183、TRPT\_Cレジスタ175、ラッチ176、デクリメンタ177、ゼロ検出回路178、比較器187、172等）、および、制御部112のIFステージ制御、命令フェッチ制御、PC部118制御等を行う部分は、IFステージ401の制御で動作する。IFステージ401は、Eステージ403のジャンプにより初期化される。

#### 【0074】

命令フェッチアドレスは、図10に示すIAレジスタ181により保持される。Eステージ403でジャンプが起こると、IAレジスタ181は、制御部112の制御のもとでJAバス323を介してジャンプ先アドレスを取り込む。この結果、IAレジスタ181は初期化される。シーケンシャルに命令データをフェッチする場合には、制御部112はインクリメンタ183にIAレジスタ181に保持されているアドレスを1インクリメントさせIAレジスタ181に書き戻す。リピート処理中で、リピートブロックの最終命令処理後リピートブロックの

先頭の命令に戻る場合、および、リピート処理終了時には、制御部 1 1 2 は I F ステージ 4 0 1 で命令処理シーケンスの切り替え制御を行う。前者の場合、P C 部 1 1 8 は、制御部 1 1 2 の制御のもとで R P T \_ S レジスタ 1 8 4 に保持されているアドレスをラッチ 1 8 5 から J A バス 3 2 3 を介して I A レジスタ 1 8 1 へ転送する。後者の場合、R P T \_ E レジスタ 1 8 6 の内容に 1 を加算したアドレスの値をラッチ 1 7 4 から J A バス 3 2 3 を介して I A レジスタ 1 8 1 へ転送する。

#### 【 0 0 7 5 】

I A レジスタ 1 8 1 に保持されている値は命令フェッチ部 1 0 2 に送られ、命令フェッチ部 1 0 2 はこの値に従い命令データをフェッチする。対応する命令データが図 8 に示す内蔵命令メモリ 1 0 3 にある場合には、命令フェッチ部 1 0 2 は内蔵命令メモリ 1 0 3 から命令コードを読み出す。この場合、1 クロックサイクルで 3 2 ビットの命令のフェッチを完了する。対応する命令データが内蔵命令メモリ 1 0 3 にない場合には、命令フェッチ部 1 0 2 は外部バスインタフェース部 1 0 6 に命令フェッチ要求を出す。外部バスインタフェース部 1 0 6 は、その命令フェッチ要求とオペランドアクセス部 1 0 4 からの要求とを調停し、命令の取り込みが可能になったら、外部メモリから命令データを取り込み、命令フェッチ部 1 0 2 に送る。外部バスインタフェース部 1 0 6 は、最小 2 クロックサイクルで外部メモリにアクセスすることが可能である。命令フェッチ部 1 0 2 は取り込んだ命令を命令キュー 1 1 1 へ転送する。

#### 【 0 0 7 6 】

命令キュー 1 1 1 は、2 エントリのキューであり、F I F O 制御で取り込んだ命令コードを制御部 1 1 2 の命令デコード部 1 1 9 へ出力する。図 1 0 に示す T R P T \_ C レジスタ 1 7 5 の内容がゼロでなく且つ命令フェッチアドレスが R P T \_ E レジスタ 1 8 6 の内容と一致したことを示すリピートブロック最終命令情報、または、T R P T \_ C レジスタ 1 7 5 がゼロであり且つ命令フェッチアドレスが R P T \_ B レジスタ 1 7 1 の内容と一致したこと（すなわち比較器 1 7 2 の一致情報が一致していること示している場合）を示すリピート処理最終命令情報が、命令キューに対応する命令コードとともに命令キュー 1 1 1 により保持され

、対応する命令コードとともに命令デコード部119に出力される。なお、RPT\_Eレジスタ186の内容とRPT\_Bレジスタ171の内容が一致する場合は、リピート処理最終命令情報が送られる。以降のステージでは、リピートブロック最終命令情報およびリピート処理最終命令情報に基づき、リピート処理に関する命令非依存のハードウェア制御が行われる。

#### 【0077】

Dステージ402において、命令デコード部119は、オペレーションコードの解析を行い、第1演算部116、第2演算部117、PC部118等で命令の実行を行うための制御信号群を生成する。Dステージ402は、Eステージ403のジャンプにより初期化される。命令キュー111から送られてくる命令コードが無効な場合には、アイドルサイクルとなり、命令デコード部119は有効な命令コードを取り込むまで待つ。Eステージ403が次の処理を開始できない場合には、命令デコード部119は第1演算部116、第2演算部117等へ送る制御信号群を無効化し、Eステージ403での先行命令の処理の終了を待つ。例えば、Eステージ403で実行中の命令がメモリアクセスを行う命令であり、Mステージ404でのメモリアクセスが終了していない場合にこのような状態になる。

#### 【0078】

Dステージ402において、命令デコード部119はシーケンシャル実行を行う2命令の分割や、2サイクル実行命令のシーケンス制御も行う。さらに、命令デコード部119は、スコアボードレジスタ（図示せず）を用いたロードオペランドの干渉チェックや第2演算部117における演算器の干渉チェック等も行い、干渉を検出した場合には、干渉が解消されるまで制御信号群の出力を抑止する。図13はロードオペランド干渉の例を示す図である。ワードまたは2ワードのロード命令の直後にロードするオペランドを参照する積和演算命令がある場合、命令デコード部119はレジスタへのロードが完了するまで積和演算命令の実行開始を抑止する。この場合、メモリアクセスが1クロックサイクルで終了する場合でも1クロックサイクルストールが起こる。バイトデータをロードする場合に、Wステージ405でレジスタファイル115への書き込みが完了するので更

に1クロックサイクルストール期間が延びる。図14は演算ハードウェア干渉の例を示す図である。例えば、積和演算命令の直後に加算器を使用する丸め命令がある場合、命令デコード部119は先行の積和演算命令の演算が終了するまで丸め命令の実行開始を抑止する。この場合1クロックサイクルストールが起こる。なお、積和演算命令が連続する場合ストールは起こらない。

#### 【0079】

第1デコーダ113は、主として第1演算部116のすべて、PC部118のIFステージ401で制御される部分以外、およびレジスタファイル115のS1バス301、S2バス302およびS3バス303への読み出し制御とD1バス311からの書き込み制御に関する制御信号群を生成する。第1デコーダ113は、また、命令に依存するMステージ404やWステージ405での処理に必要な制御信号群も生成し、パイプラインの処理の流れに付随してMステージ404やWステージ405へ転送する。一方、第2デコーダ114は、主として第2演算部117での実行制御、レジスタファイル115のS4バス304およびS5バス305への読み出し制御とD2バス312およびD3バス313からの書き込み制御に関する制御信号群を生成する。

#### 【0080】

命令デコード部119は、また、命令キュー111から取り込まれたリピートブロック最終命令情報およびリピート処理最終命令情報に基づき、命令に依存しないリピート処理に関する図10に示すNPC191の更新に関する制御信号、RPT\_Cレジスタ188の更新に関する制御信号や、図9のPSWラッチ222のRPビット43の更新に関する制御信号などを生成する。

#### 【0081】

Eステージ403では、演算、比較、制御レジスタを含むレジスタ間転送、ロード／ストア命令のオペランドアドレス計算、ジャンプ命令のジャンプ先アドレスの計算、ジャンプ処理、EIT（例外、割り込み、トラップの総称）検出と各EITのベクタアドレスへのジャンプ等、メモリアメセスと積和／積差演算命令の加算処理を除く命令実行に関するほとんどすべての処理が行われる。

#### 【0082】

割り込みイネーブルの場合の割り込みの検出は、必ず32ビット命令の切れ目で行われる。従って、32ビット命令の中にシーケンシャルに実行される2つのショート命令がある場合、この2つのショート命令間でいかなる割り込みも受け付けられない。

#### 【0083】

Eステージ403で処理中の命令がオペランドアクセスを行う命令であり、Mステージ404でメモリアクセスが完了していない場合には、Eステージ403での完了は待たされる。この際、Eステージ403の制御は制御部112で行われる。

#### 【0084】

Eステージ403において、第1演算部116のALU153は、算術論理演算、比較、転送を行う。第1演算部116のALU153は、また、モジュロの制御を含むメモリオペランドのアドレス計算や分岐先のアドレス計算も行う。レジスタファイル115は、オペランドとして指定されたレジスタに保持されている値をS1バス301、S2バス302およびS3バス303に出力する。ALU153は、演算やアドレス計算等を行う場合、これらのバスに出力された値や、必要に応じて別途取り込まれる即値データ、変位等の拡張データを使用して必要な演算を行う。汎用レジスタへの書き込みを行う場合、演算結果はD1バス311を介してレジスタファイル115中のオペランドにより指定されたレジスタに書き込まれる。ロード/ストア命令の場合には、情報処理装置は、オペランドアドレスとなるALU153の出力をAOラッチ154およびOAバス321を介してオペランドアクセス部104へ転送する。ジャンプ命令の場合には、情報処理装置は、ジャンプ先アドレスとなるALU153の出力をJAバス323を介してPC部118へ転送する。第1演算部116は、レジスタファイル115からS1バス301およびS2バス302上に出力されたストアデータをSDレジスタ160およびラッチ161を介して整置回路162へ転送する。そして、整置回路162は受け取ったストアデータを整置する。PC部118は、実行中の命令のPC値の管理、次に実行する命令のアドレスの生成を行う。第1演算部116およびPC部118に含まれる制御レジスタ（アキュムレータを除く）と

レジスタファイル 1 1 5 と間でのデータの転送は、S 3 バス 3 0 3 および D 1 バス 3 1 1 を介して行われる。

#### 【 0 0 8 5 】

E ステージ 4 0 3 において、第 2 演算部 1 1 6 は、算術論理演算、比較、転送、シフト他積和演算の加算以外のすべての演算を行う。第 2 演算部 1 1 6 の各演算器は、レジスタファイル 1 1 5、即値レジスタ 2 1 8、アキュムレータ 2 0 8 等から、S 4 バス 3 0 4 および S 5 バス 3 0 5 やその他の専用経路を介してオペランドの値を受け取ると、指定された演算を行う。演算結果は、アキュムレータ 2 0 8、または、D 2 バス 3 1 2 および D 3 バス 3 1 3 を介してレジスタファイル 1 1 5 に書き戻される。

#### 【 0 0 8 6 】

第 1 演算部 1 1 6 および第 2 演算部 1 1 7 による演算結果に基づいた P S W ラッチ 2 2 2 に保持された値の更新制御も E ステージ 4 0 3 で行われる。しかし、演算結果の確定が E ステージ 4 0 3 の遅い時期に行われるので、P S W ラッチ 2 2 2 に保持されている値の更新は実際には次のクロックサイクルで行われる。一方、データ転送による P S W ラッチ 2 2 2 により保持されている値の更新は対応するクロックサイクルで完了する。

#### 【 0 0 8 7 】

E ステージ 4 0 3 では、実行される命令に依存しない P C 値の更新やリピート制御も行われる。新たに 3 2 ビット命令の処理を開始するたびに、P C 部 1 1 8 は、制御部 1 1 2 の制御のもとでラッチ 1 9 2 に保持されている値を E P C 1 9 4 へ転送する。N P C 1 9 1 は次に処理する命令のアドレスを保持する。E ステージでジャンプが起こった場合、情報処理装置は、第 1 演算部 1 1 6 の A L U 1 5 3 により生成されたジャンプ先アドレスを J A バス 3 2 3 を介して N P C 1 9 1 へ書き込み、N P C 1 9 1 を初期化する。シーケンシャルに命令を継続的に処理する場合には、3 2 ビット命令の処理を開始するたびに、P C 部 1 1 8 は、制御部 1 1 2 の制御のもとでインクリメント 1 9 3 により 1 インクリメントされた値を N P C 1 9 1 へ書き戻す。リピートブロック最終命令の処理を開始する際には、N P C 1 9 1 はラッチ 1 8 5 からリピートブロックの先頭アドレスを取り込

む。リピート処理最終命令の処理を開始する際には、NPC191はラッチ174からリピートブロックの次の命令のアドレスを取り込む。そして、リピートブロック最終命令の処理を終了するクロックサイクルにおいて、PC部118は、制御部112の制御のもとでIFステージ401ですでに1デクリメントされているカウンタ値をラッチ176からRPT\_Cレジスタ188へ転送する。リピート処理最終命令の処理を終了するクロックサイクルにおいて、制御部112はPSWラッチ222のRPビット43をゼロクリアする。

#### 【0088】

第2デコーダ114により生成された積和／積差演算の加減算実行のための制御信号群は、Eステージ403制御のもとで保持される。メモリアクセス情報およびロードレジスタ情報は、Mステージ404に送られる。制御部112はEステージ403のステージ制御も行う。

#### 【0089】

Mステージ404では、第1演算部116から送られたアドレスでオペランドのアクセスが行われる。オペランドアクセス部104は、オペランドが内蔵データメモリ105やチップ内I/O（図示せず）にある場合には、内蔵データメモリ105やチップ内I/Oに対し、1クロックサイクルに1回のオペランドのリードまたはライトを行う。オペランドが内蔵データメモリ105やチップ内I/Oにない場合には、オペランドアクセス部104は外部バスインタフェース部106へデータアクセス要求を出す。外部バスインタフェース部106は、外部メモリに対してデータアクセスを行い、ロードの場合には読み出されたデータをオペランドアクセス部104へ転送する。外部バスインタフェース部106は、最小2クロックサイクルで外部メモリへのアクセスを行うことが可能である。ロードの場合には、オペランドアクセス部104は読み出したデータをODバス322に出力する。バイトデータロードの場合、LDレジスタ164がODバス322上のロードデータを取り込む。ワードまたはダブルワードデータロードの場合、レジスタファイル115がODバス322上のロードデータを直接取り込む。ストアの場合には、第1演算部116は、制御部112の制御のもとで整置されたストアデータの値を整置回路162からラッチ163およびODバス322を



介してオペランドアクセス部104へ転送し、対象となるメモリへの書き込みが行われる。制御部112はMステージ404のステージ制御も行う。

#### 【0090】

Wステージ405において、第1演算部116は、LDレジスタ164に保持されたロードオペランド（バイト）をラッチ165へ転送し、整置回路166により整置、ゼロ／符号拡張した後にWバス314に出力する。レジスタファイル115はWバス314上のロードデータを命令で指定されたレジスタに書き込む。

#### 【0091】

E2ステージ406において、図11に示す第2演算部117のALU201は、積和／積差演算の加減算処理を行い、加減算結果をアキュムレータ208に書き戻す。

#### 【0092】

この実施の形態1による情報処理装置は入力クロックに基づいて内部制御を行う。最短の場合、各パイプラインステージでは、1クロックサイクルで処理が終了する。なお、クロック制御はこの発明に直接関係ないので以下ではその説明を省略する。

#### 【0093】

次にサブ命令の処理例について説明する。

加減算、論理演算、比較等の演算命令やレジスタ間の転送命令を実行する場合、IFステージ401、Dステージ402、Eステージ403の3段でその命令の処理が終了する。その際、演算やデータ転送はEステージ403において行われる。積和／積差命令を実行する場合、乗算を行うEステージ403および加減算を行うE2ステージ406がそれぞれ1クロックサイクルで処理される。すなわち、4段で積和／積差命令の処理が終了する。

#### 【0094】

バイトロード命令を実行する場合、IFステージ401、Dステージ402、Eステージ403、Mステージ404、Wステージ405の5段でその命令の処理が終了する。ワード／2ワードロードやストア命令を実行する場合、IFステ

ージ401、Dステージ402、Eステージ403、Mステージ404の4段でその命令の処理が終了する。

#### 【0095】

非整置アクセスの場合には、オペランドアクセス部104が、Mステージ404の制御のもと非整置アクセスを2回のアクセスに分割しメモリアksesを行う。また、実行に2クロックサイクルかかる命令では、第1および第2命令デコーダ113、114が、それぞれ各クロックサイクル毎に制御信号群を生成し、2クロックサイクルかけて演算実行を行う。

#### 【0096】

1つの32ビット命令が1つのロング命令で構成されている場合、この1つのロング命令を実行することにより32ビット命令の実行が完了する。また、平行に2つのショート命令を実行する場合、これらの2つのショート命令を実行するのに要するクロックサイクル数は、2つのショート命令のうち処理サイクル数の大きい方の命令により決まる。例えば、2クロックサイクル実行の命令と1クロックサイクル実行の命令との組み合わせの場合、これらの命令を平行に実行するには2クロックサイクル要する。2つのショート命令をシーケンシャルに実行する場合、これら2つのショート命令がシーケンシャルにデコードされ実行される。例えば、Eステージ403において1クロックサイクルで実行が完了する加算命令を2つシーケンシャルに実行する場合、Dステージ402、Eステージ403とも各加算命令の処理に1クロックサイクル要し、各加算命令の実行には合計2クロックサイクル要する。Eステージ403における先行命令の実行と並列して、Dステージ402で後続命令のデコードが行われる。

#### 【0097】

次にこの実施の形態1による情報処理装置のリピート処理の動作を詳細に説明する。

図15はリピート命令“REP Rsrc1, Rsrc2, displ6”のビット割り付けを示す図である。リピート命令は図6に示すロング命令のフォーマットを有している。図15において、501はFMビットであり、ここでは、ロングフォーマットを示す“11”である。また、502はオペレーションコー

ドであり、503はリピートブロック内のリピート処理が終了されるリピートブレイクアドレスを指定するレジスタ番号Rsrc1である。なお、リピートブレイクアドレスは、リピートブロックの開始命令アドレスからの変位で指定され、Rsrc1にはこの変位値が設定される。リピートブロック内で分岐等が起こらない場合には、Rsrc1に設定される値は“最終リピートサイクルで実行されるリピートブロック内の命令数-1”に等しい。リピートブレイクアドレスとしてリピートブロック内の任意の命令のアドレス（ただし開始命令アドレスからの変位で）を指定可能である。リピートブロックの開始命令や最終命令のアドレスであっても構わない。また、504は（リピートブロックの繰り返し回数（途中でブレイクする最後の回も含む）-1）を保持するレジスタを指定するレジスタ番号Rsrc2であり、505はリピートブロック最終命令のリピート命令からの変位値disp16である。従って、リピートブロックは、リピート命令の直後の命令からdisp16で指定される命令までの範囲に等しい。この実施の形態1では、パイプライン処理の都合上、リピートブロックが32ビット命令で2命令以下であるか、または、レジスタ番号Rsrc2により指定されるレジスタに設定される値が“0”でレジスタ番号Rsrc1により指定されるレジスタに設定される値が“2”以下の場合には、動作保証しないものとする。

【0098】

次に以下の積和演算を例に具体的な動作について説明する。

【0099】

【数1】

$$\sum_{i=0}^{N-1} (A[i] * B[i]) \quad \dots (1)$$

【0100】

図16は上記式(1)に従って積和演算を行うアセンブラ記述の、REP命令を含むプログラム例を示す図である。なお、図において、各行のセミコロン以降はコメントである。また、“||”は2つのショート命令を並列に実行することを示している。

【0101】

以下では、汎用レジスタR8 9にはA[0]のアドレスが、汎用レジスタR9 10にはB[0]のアドレスが、汎用レジスタR10 11には“積和演算回数N”が設定されているものとする。また、A[i]およびB[i] (i=0~N-1)は16ビットデータであり、A[0]~A[N-1]およびB[0]~B[N-1]はともにこの順番で内蔵データメモリ105上のアドレスが順に増大する複数のメモリ位置にそれぞれ格納されており、A[0]およびB[0]は、32ビット(4バイト)整置されているものとする。また、積和演算結果は16ビットに丸められ、汎用レジスタR0 1に保持されるものとする。

#### 【0102】

なお、以下では、図16に示す10個の32ビット命令をそれぞれI1~I10で参照し、並列に実行される2命令における“||”の左側の命令をIx a (xは2~4または6~10のいずれか)、“||”の右側の命令をIx b (xは2~4または6~10のいずれか)で参照する。例えば、I2の命令において、LD2W命令をI2 a、SRLI命令をI2 bで参照する。I1からI4の命令はリピート処理を行うための前処理であり、I5の命令はリピート命令であり、I6からI9の命令は積和演算を行うためのリピートブロックであり、I10の命令は後処理である。

#### 【0103】

この実施の形態1による情報処理装置は、2つの異なる領域に割り当てられた配列データをロードする場合、異なるロード命令で読み出す必要がある。また、Mステージ404でロードが行われるので、パイプラインストールなしに積和演算を実行するためには、積和命令で参照するオペランドデータは、内蔵データメモリ105に格納されている場合でも2クロックサイクル以上前に実行される命令でロードされる必要がある。従って、1クロックサイクルに1回のスループットで積和演算を実行するためには、最低4命令でリピート処理を行う必要がある。汎用レジスタR0~R3 1~4の4つのレジスタがA[i]のバッファとして、汎用レジスタR4~R7 5~8の4つのレジスタがB[i]のバッファとして使用される。

#### 【0104】

I 1 の AND 3 命令を実行する場合、情報処理装置は、汎用レジスタ R 1 0 1 1 に保持されているデータと即値データ “3” のビット毎との論理積をとり、その結果を汎用レジスタ R 1 1 1 2 に書き込む。図 1 6 の I 5 に示すように、汎用レジスタ R 1 1 に保持されている値がリピート命令のレジスタ番号 R s r c 1 で指定される値である。I 2 b の S R L I 命令は、汎用レジスタ R 1 0 1 1 に保持されている値を 2 ビット算術右シフトする命令である。汎用レジスタ R 1 0 1 1 に保持されている値がリピート命令のレジスタ番号 R s r c 2 で指定される値となる。例えば、上記式 (1) における N が 1 0 の場合、汎用レジスタ R 1 0 1 1 に設定される初期値は “9” であり、上記 I 1 および I 2 の命令の実行により、汎用レジスタ R 1 0 1 1 および汎用レジスタ R 1 1 1 2 にはそれぞれ “2” と “1” が書き込まれる。I 3 b の N O P 命令はノーオペレーション命令である。この命令は、命令コードの整置のために挿入されており、実行されても有効なオペレーションは行われぬ。I 4 b の C L R A C 命令は、アキュムレータ A 0 3 1 をゼロクリアする命令である。

#### 【0105】

リピート処理を開始するために、前処理として、オペランドデータのロードを行っておく必要がある。I 2 a, I 3 a および I 4 a の L D 2 W 命令がその処理を行う。たとえば、I 2 a の L D 2 W 命令は、ポストインクリメント付きレジスタ間接モードの 2 ワードロード命令である。この命令が実行されると、汎用レジスタ R 8 9 で指定されるアドレスから始まる 1 6 ビットデータ 2 つが汎用レジスタ R 0 1 および汎用レジスタ R 1 2 にそれぞれロードされ、汎用レジスタ R 8 9 に保持されている値がオペランドサイズに相当する 4 だけポストインクリメントされる。

#### 【0106】

情報処理装置は次に I 5 の R E P 命令を実行する。R E P 命令は 2 クロックサイクルかけて実行される。情報処理装置は、第 1 クロックサイクルで図 1 に示す制御レジスタ C R 9 2 8 (R P T \_ E)、制御レジスタ C R 7 2 6 (R P T \_ C) にそれぞれ相当する R P T \_ E レジスタ 1 8 6、R P T \_ C レジスタ 1 8 8 等の設定を行い、第 2 クロックサイクルで図 1 に示す制御レジスタ C R 6 2

5 (RPT\_\_B)、制御レジスタCR8 27 (RPT\_\_S) にそれぞれ相当するRPT\_\_Bレジスタ171、RPT\_\_Sレジスタ184等の設定を行い、制御レジスタCR0 21のPSWのRPビット43を“1”に設定する。

#### 【0107】

まず、第1クロックサイクルにおいて、PC部118は、制御部112の制御のもとでREP命令のPC値をEPC194からS3バス303に出力し、第1演算部116のAAラッチ151はS3バス303上に出力されたそのPC値を取り込む。また、ABラッチ152は、第1デコーダ113から出力されるREP命令のdisp16で指定される変位値を取り込む。第1演算部116のALU153は、AAラッチ151およびABラッチ152に保持されている値同士を加算し、リピートブロックの最終命令のアドレスであるI9の命令のアドレスを計算する。第1演算部116は制御部112の制御のもとでALU153による計算結果をJAバス323に出力し、PC部118のRPT\_\_Eレジスタ186はJAバス323上に出力されたその計算結果を取り込む。その後、PC部118のインクリメンタ173は、RPT\_\_Eレジスタ186に保持されている値を1インクリメントする。そのインクリメント結果はラッチ174に書き込まれる。その結果、ラッチ174は、リピートブロックの次の命令すなわちI10の命令のアドレスを保持する。また、上記処理と並行して、I5のREP命令のレジスタ番号Rsrc2で指定される値すなわち汎用レジスタR10 11に保持されている値が、S1バス301を介して第1演算部116のラッチ169へ転送され、さらに、セクタ155およびD1バス311を介してRPT\_\_Cレジスタ188およびTRPT\_\_Cレジスタ175の両方に書き込まれる。そして、RPT\_\_Cレジスタ188に書き込まれた値はさらにラッチ179へ転送される。

#### 【0108】

第2クロックサイクルでは、PC部118のNPC191がリピートブロックの開始命令のアドレスであるI6の命令のアドレスを保持している。PC部118は制御部112の制御のもとでNPC191に保持されている値をS3バス303に出力し、第1演算部116のABラッチ152はS3バス303上に出力

されたその値を取り込む。また、REP命令のレジスタ番号Rsrc1により指定される汎用レジスタR11 12に保持されている値が、S2バス302を介して第1演算部116のAAラッチ151へ転送される。そして、ALU153は、AAラッチ151およびABラッチ152に保持されている値同士を加算し、リピート処理の最後に実行される命令のアドレスを計算する。第1演算部116は制御部112の制御のもとでALU153による計算結果をJAバス323に出力し、PC部118のRPT\_Bレジスタ171はJAバス323上に出力されたその計算結果を取り込む。また、上記処理と並行して、PC部118は、NPC191に保持されている値をラッチ192およびD1バス311を介してRPT\_Sレジスタ184へ転送する。その後、PC部118は、さらに、RPT\_Sレジスタ184に保持されている値をラッチ185へ転送する。そして、制御部112はPSWラッチ222のRPビット43を“1”に設定する。

## 【0109】

図17および図18はこの実施の形態1による、図16に示したプログラムのREP命令実行からリピート処理完了までのパイプライン処理を示すタイミングチャート図である。これら2つの図は連続した図であり、わかりやすいように2クロックサイクルT8, T9は重複して示している。図の上部4行は各パイプラインステージの処理を示しており、残りの下部はリピート処理に関するレジスタに保持されている値や信号の値を示している。また、簡単のため、各命令およびアドレスをI5等命令の仮称で示している。“IA==RPT\_E”、“IA==RPT\_B”はそれぞれ、比較器187, 172の比較結果を示しており、“TRPT\_C==0”はゼロ検出回路178の出力を示している。また、図17および図18は10回の積和を行う場合のパイプライン処理を示している。図示した例は、リピート処理最終命令がI7の命令であり、I6~I9の4命令のリピートブロックの処理を2回さらにI6, I7の2命令を実行した後、I10の命令の実行を行うものである。

## 【0110】

図17に示すように、情報処理装置は、2クロックサイクルT3, T4でI5のREP命令を実行する。上記したようにI6~I9までの4命令がリピートブ

ロックであり、情報処理装置はこれらの命令を繰り返し実行する。情報処理装置は2ワードのロード(LD2W)命令と積和演算(MAC)命令とを並列に実行する。例えば、I6bのMAC命令は、汎用レジスタR0 1に保持されている値と汎用レジスタR4 5に保持されている値とを乗算し、乗算結果とアキュムレータA0 31に保持されている値とを加算して、アキュムレータA0 31に書き戻すものである。ロードオペランドの干渉を起こさず1クロックサイクルで1回の積和を実行できるように、I6, I7の命令で参照されるデータはI7, I8の命令で、I8, I9の命令で参照されるデータはI9, I6の命令でロードされる。積和演算自体の詳細はこの発明とは直接関連がないので、以下ではその説明を省略する。

#### 【0111】

次に実行する命令とは無関係にハードウェア的に行われるリピート処理中の命令処理シーケンス制御について詳細に説明する。

#### 【0112】

実行する命令処理シーケンスの切り替えはIFステージ401で行われる。この際、PC部118の比較器187は、IAレジスタ181に保持されている命令フェッチアドレスとRPT\_Eレジスタ186に保持されているリピートブロック最終命令のアドレスとを比較する。また、比較器172は、IAレジスタ181に保持されている命令フェッチアドレスとRPT\_Bレジスタ171に保持されているリピート処理の最終実行命令のアドレスとを比較する。

#### 【0113】

PSWラッチ222のRPビット43が“1”(リピート処理中)に設定されており且つTRPT\_Cレジスタ175に保持されている値が非ゼロである状態でI9の命令をフェッチした場合、比較器187による比較結果は比較された値同士が一致していることを示しており、これを受けて制御部112はリピートブロックの先頭のI6の命令を次に実行すると判断する。図17のクロックサイクルT6および図18のクロックサイクルT10がこの状態に対応する。I9の命令のアドレス出力完了後、ラッチ185に保持されているリピートブロックの先頭命令であるI6の命令のアドレスがJAバス323を介してIAレジスタ18



1へ転送されると、命令フェッチ部102は次にI6の命令をフェッチする。また、2クロックサイクルT6, T10では、デクリメンタ177がTRPT\_Cレジスタ175に保持されている値を1デクリメントし、デクリメント結果はTRPT\_Cレジスタ175に書き戻される。この間、TRPT\_Cレジスタ175に保持されている値が非ゼロであるならば、制御部112は比較器172の比較結果を無視する。

## 【0114】

リピートブロック最終命令情報がI9の命令の実行に同期してパイプライン中で転送され、この情報に基づき以下の処理が行われる。Eステージ403でのI9の命令の実行開始前(2クロックサイクルT7, T11)、PC部118は次に実行する命令のPC値としてI6の命令のアドレスをラッチ185からNPC191へ転送する。また、Eステージ403でI9の命令処理を完了する2クロックサイクルT8, T12で、PC部118はTRPT\_Cレジスタ175に保持されている値をラッチ176を介してRPT\_Cレジスタ188およびラッチ179へ転送する。従って、RPT\_Cレジスタ188は、常にEステージ403で実行命令に連動した状態更新情報を保持する。

## 【0115】

PSWラッチ222のRPビット43が1(リピート処理中)であり且つTRPT\_Cレジスタ175に保持されている値がゼロである状態で、I7の命令をフェッチした場合、比較器172による比較結果は比較された値同士が一致していることを示している。図18のクロックサイクルT12がこの状態に対応する。I7の命令アドレス出力完了後、PC部118が、制御部112の制御のもとでラッチ174に保持されているリピートブロックの次の命令であるI10の命令のアドレスをJAバス323を介してIAレジスタ181へ転送すると、命令フェッチ部102は次にI10の命令をフェッチする。この場合、デクリメンタ177はTRPT\_Cレジスタ175に保持されている値をデクリメントしない。

## 【0116】

リピート処理最終命令情報はI7の命令の実行に同期してパイプライン中で転

送され、この情報に基づき以下の処理が行われる。Eステージ403でのI7の命令の実行開始前（クロックサイクルT13）、PC部118は、次に実行する命令のPC値としてI10の命令のアドレスをラッチ174からNPC191へ転送する。また、Eステージ403でI7の命令処理を完了するクロックサイクルT14で、制御部112はPSWラッチ222のRPビット43をゼロクリアする。これで、リピート処理は完了したことになる。

#### 【0117】

リピート処理終了後、リピートブロックの次の命令（I10）の処理が行われる。I10aのRACHI命令は、アキュムレータA031に保持されている値のビット8～23（A0H）を16ビットに丸め、サチュレーション処理を行い、その結果を汎用レジスタR01に書き込む命令である。この例では、I7bのMAC命令とI10のRACHI命令とは演算ハードウェア干渉を起こすので、クロックサイクルT15でのI7の命令のEステージ406における処理終了後クロックサイクルT16でI10の命令が実行される。

#### 【0118】

以上リピート処理最終命令がI7の命令である例を示したが、リピート処理最終命令はこれに限定されるものではなくリピートブロック内のどの命令であっても良い。たとえば、リピート処理最終命令はリピートブロックの開始命令I6であってもよいし、リピートブロックの最終命令I9であってもよい。さらに、リピート処理最終命令はリピートブロック以外の命令であっても良く、この命令は例えばリピートブロック内でサブルーチンコールなどを行う場合のサブルーチン内の命令であり得る。

#### 【0119】

なお、ロードに関しては、3命令分投機的に発行され実行される。

リピート処理中に、Eステージ403でジャンプ命令が実行された場合には、リピート関係の前処理に優先してジャンプ処理が行われる。この際、第1演算部116のALU153は生成したジャンプ先アドレスをJAバス323に出力し、PC部118のNPC191およびIAレジスタ181はJAバス323上に出力されたそのジャンプ先アドレスを取り込む。また、実行ステージジャンプ時

には、PC部118はラッチ179に保持されている値をTRPT\_Cレジスタ175へ転送し、TRPT\_Cレジスタ175に保持されていた先行更新情報をクリアする。ただし、リピートブロックの最終命令でジャンプが起こった場合には、更新された情報をRPT\_Cレジスタ188に反映させるため、PC部118はTRPT\_Cレジスタ175へのラッチ179に保持されている値の転送を行わない。

#### 【0120】

次にリピート処理中に外部割り込みを受け付けた場合の処理について簡単に説明する。

#### 【0121】

この実施の形態1による情報処理装置は、32ビット命令の切れ目で外部割り込みを受け付けることができる。制御レジスタCR0 21に保持されているPSWのIEビット42が“1”の状態では外部割り込みを検出すると、情報処理装置は割り込みを受け付ける。割り込みを受け付けると、Eステージ403はハードウェア的に1クロックサイクルの間ノーオペレーション状態となり、情報処理装置は、ハードウェア的に生成される制御信号に基づいて1クロックサイクル割り込み処理を行う。まず、PC部118は、割り込み処理の前クロックサイクルにおいて、NPC191に保持されている次に実行する命令のPC値をラッチ192を介してEPC194へ転送する。これは、割り込み処理復帰後実行する命令のアドレスをEPC194に保持されている値に戻すためである。割り込み処理サイクルでは以下の処理を行う。PC部118は、EPC194に保持されている値をラッチ195を介してBPC196に待避する。また、情報処理装置は、PSWラッチ222に保持されている値をD1バス311を介して第1演算部116のBPSWレジスタ168に待避する。その後、制御部112のPSW部221は、PSWラッチ222をハードウェア的にゼロクリアする。そして、情報処理装置は、仕様としてあらかじめ決まっている割り込み処理で最初に実行する命令のアドレスをハードウェア的に生成し、第1演算部116のABラッチ152へ転送する。第1演算部116のALU153は、ABラッチ152に保持されている値と0とを加算し、加算結果をJAバス323に出力する。PC部1

18は、JAバス323上に出力されたその加算結果をNPC191およびIAレジスタ181に取り込み、ジャンプ処理を行う。割り込み処理ではPSWラッチ222のRPビット43がゼロクリアされるので割り込み前に行われていたリピート処理は中断される。第1演算部116のBPSWレジスタ168は、割り込み前にリピート処理中であった旨を示す情報を保持している。

#### 【0122】

この実施の形態1による情報処理装置は、図1に示す制御レジスタと汎用レジスタと間の転送を行う命令を備えており、割り込み処理を行う前に必要に応じて割り込み処理ハンドラ内で、リピート関連の制御レジスタに保持されている値をコンテキスト情報としてメモリに待避し、割り込み処理終了後に制御レジスタの内容を割り込み処理以前の状態に復帰させることができる。制御レジスタに保持されている値を汎用レジスタへ転送する場合には、制御レジスタからS3バス303上に値が出力され、ABラッチ152、ALU153（ゼロとの加算を行う）、セクタ155およびD1バス311を介してレジスタファイル115へ転送される。汎用レジスタから制御レジスタへ転送する場合には、レジスタファイル115からS3バス303に復帰される値が出力され、ABラッチ152、ALU153（ゼロとの加算を行う）、セクタ155およびD1バス311を介して、各制御レジスタに書き込まれる。REP命令実行時と同様、リピート関連の制御レジスタ更新時には、リピート制御に必要な関連リソースの更新も行われる。例えば、RPT\_Cレジスタ188への書き込み時には、TRPT\_Cレジスタ175へも同時に書き込まれる。また、RPT\_Sレジスタ184やRPT\_Eレジスタ186の更新時には、ラッチ185、174に保持されている値も連動して更新される。

#### 【0123】

割り込み処理から復帰する場合は、情報処理装置は割り込み処理前に待避したコンテキスト情報を汎用レジスタ、制御レジスタ等へ復帰させた後、割り込み処理からのリターン（RTE）命令を実行する。以下では、割り込み受け付け前にリピート処理中であった場合を考える。RTE命令実行前に、情報処理装置は、第1演算部116のBPSWレジスタ168、PC部118のBPC196、お

よびリピート関連の制御レジスタ等に保持されている値を割り込み処理起動直後の状態に復帰させる。この結果、B P S W レジスタ 1 6 8 の P S W の R P ビット 4 3 に対応するビットは“1”になる。この状態で、情報処理装置は R T E 命令を実行する。R T E 命令の実行では以下の処理が行われる。P C 部 1 1 8 は B P C 1 9 6 に保持されている値を S 3 バス 3 0 3 に出力し、第 1 演算部 1 1 6 の A B ラッチ 1 5 2 は S 3 バス 3 0 3 上に出力されたその値を取り込む。A L U 1 5 3 は、A B ラッチ 1 5 2 に保持されている値とゼロとを加算し、加算結果を J A バス 3 2 3 に出力する。P C 部 1 1 8 は J A バス 3 2 3 上に出力されたその加算結果を N P C 1 9 1 および I A レジスタ 1 8 1 に取り込む。その結果、情報処理装置は B P C 1 9 6 に保持されていたアドレス値へのジャンプ処理を行う。また、第 1 演算部 1 1 6 は、B P S W レジスタ 1 6 8 に保持されている値を T P S W ラッチ 1 6 7 を介して制御部 1 1 2 へ転送し、P S W ラッチ 2 2 2 はその値を取り込む。この結果、P S W ラッチ 2 2 2 の R P ビット 4 3 に 1 がセットされ、リピート処理が再開される。

## 【 0 1 2 4 】

このように、この実施の形態 1 による情報処理装置は、リピート中に割り込み、トラップまたは例外処理が起動されても、復帰後正しく中断されていたリピート処理を再開できる。

## 【 0 1 2 5 】

上述の R E P 命令は、リピート関連のパラメータ設定を 1 命令で実行できるので非常に使いやすいが、その反面、実行ステージで各種リソースの設定を行うので、R E P 命令実行時にすでにフェッチを終了してしまう命令に関しては、命令フェッチ段階でのシーケンス制御ができないという欠点がある。従って、R E P 命令使用時のリピートブロックの最小サイズや繰り返し回数の最小値はパイプラインの構成や命令バッファ（命令キュー）のサイズなどに依存しており、リピートブロックの最小サイズや繰り返し回数の最小値には制限がある。例えば、上述の R E P 命令では、上記したように、リピートブロックが 2 つ以下の 3 2 ビット命令から成る場合、または、レジスタ番号 R s r c 2 により指定される値が“0”でレジスタ番号 R s r c 1 により指定される値が“2”以下の場合には、動作

保証しないものとしている。

【0126】

次に上述のものとは異なるリピート処理の起動方法について簡単に説明する。

この実施の形態1による情報処理装置は、図2に示すPSWのRPビット43をセットするSETRP命令を備えている。図19はリピートブロック内の1命令しか実行しない場合の処理例を示すタイミングチャート図である。図19は、I5の命令がSETRP命令であり、I6～I9の命令がリピートブロックであり、I6の命令がリピート処理最終命令である場合を示している。SETRP命令実行前に、情報処理装置は、図1に示すRPT\_\_S, RPT\_\_E, RPT\_\_C, RPT\_\_Bをそれぞれ対応する制御レジスタ等へ初期設定する転送命令を実行する。なお、RPT\_\_Cはゼロに設定される。すなわち、情報処理装置は、PC部118のRPT\_\_Cレジスタ188およびTRPT\_\_Cレジスタ175にゼロを書き込む。

【0127】

命令デコード部119は、IFステージ401で命令がフェッチされた段階でその命令のプリデコードを行い、その命令がSETRP命令であるか否かを判別する。そして、SETRP命令であるならばテンポラリリピート（TRP）ビットを“1”にセットする。TRPビットは、制御部112内のラッチ（図示せず）に保持されており、Eステージ403に先行してIFステージ401で管理される情報であり、PSWのRPビット43に対応している。このTRPビットによって、IFステージによりリピート処理が行われるかどうかは制御される。

【0128】

TRPビットが“1”の状態になると、上記のREP命令と同様のハードウェアリピート（シーケンス）制御が行われる。TRPT\_\_Cレジスタ175に保持されている値がゼロであり、RPT\_\_Bレジスタ171に保持されている値とIAレジスタ181に保持されている値とが一致した場合には、制御部112はTRPビットをゼロクリアしリピート処理を終了する。

【0129】

Eステージ403で管理しているPSWラッチ222のRPビット43は、S

E T R P 命令が E ステージ 4 0 3 で処理されるクロックサイクル T 5 でセットされる。E ステージ 4 0 3 でジャンプが起こった場合には、実行ステージで管理している P S W ラッチ 2 2 2 の R P ビット 4 3 の値が T R P ビットに書き込まれ、E ステージ 4 0 3 での更新に先立ち I F ステージ 4 0 1 で更新され実行されなかった情報が取り消される。

#### 【 0 1 3 0 】

S E T R P 命令を簡単に判別できるように命令のエンコードをしておけば、プリデコードに要する時間を短縮できる。また、パイプライン処理上の制約に応じて、リピート関連の制御レジスタの設定を S E T R P 命令の何命令前に設定完了しておけば良いかを規定すれば良い。この実施の形態 1 では、S E T R P 命令の 2 命令前までにリピート関連の制御レジスタの設定を完了しておく必要がある。また、S E T R P 命令は R E P 命令と異なりフラグを設定するだけなので、必ずしもリピートブロックの直前に実行する必要はない。パイプライン処理としてタイミングが間に合うように、リピートブロックの 2 命令以上前に実行するように制約を設けても良い。ただし、設定をしてから条件判定によりリピートブロックの実行を行わない場合については、S E T R P 命令の使用を禁止する等別途制約を設ける必要がある。

#### 【 0 1 3 1 】

R E P 命令と異なり、S E T R P 命令では、図 1 に示す R P T \_ C, R P T \_ S, R P T \_ E および R P T \_ B の 4 つの値を先行する命令で対応する制御レジスタに初期設定しておく必要がある。ただし、同一ルーチンを繰り返し呼び出す場合には、R P T \_ S, R P T \_ E および R P T \_ B は毎回設定する必要はない。

#### 【 0 1 3 2 】

上記したように、この実施の形態 1 による情報処理装置は、実行される命令で特定されるオペレーションとは独立して、リピートブロック内でリピート処理の終了を検出し、検出するとリピートブロックの途中でリピート処理を終了し、リピートブロックの次の命令に命令処理シーケンスを切り替える、ハードウェアにより実現した機能を備えている。従って、リピートブロックを繰り返し処理する

繰り返し回数がダイナミックに変わり且つリピートブロックの途中でリピート処理を終了する場合にも、同一のプログラムで高速にリピートブロックを所定の繰り返し回数処理できる。例えば、上述のREP命令を用いた積和演算の例で4回以上の積和を行う場合でも、リピートブロックの処理を途中で終了するハードウェアにより実現された機能を有していないならば、ロードの投機実行を許しても、繰り返し回数が $4n$ ,  $4n+1$ ,  $4n+2$ ,  $4n+3$  ( $n$ は1以上の整数)の少なくとも4通りのいずれかであるかを条件判定する必要がある。これら4つの場合についてそれぞれプログラムが必要になる。また、3回以下の積和にも対応するために、それに対応した条件判定およびプログラムが必要になる。積和演算等の処理に関して演算ライブラリを準備する場合においても、繰り返し回数をパラメータとすると同様なことがいえる。

#### 【0133】

また、繰り返し回数が静的に設定される場合にも、ロード処理の投機実行を行ってもよいならば、例えば、上記式(1)に基づき4回以上積和演算する図16に示すようなプログラムを作成することが可能であり、繰り返し回数がダイナミックに指定される場合と同様の効果を得ることができる。すなわち、コード効率を上げると共にプログラムを簡素化してプログラムの開発効率を上げることが可能である。

#### 【0134】

また、この実施の形態1による情報処理装置は、リピートブロックが複数命令で構成される場合にも、オーバーヘッドなくリピートブロックの途中でリピート処理を終了できるので、上記の繰り返し回数の判定が不要となり、且つ、1つのプログラムで任意回数の繰り返しに対処できる。従って、条件判定、分岐のための前処理/後処理によるオーバーヘッドがなくなり性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる。特に、デジタル信号処理においては、繰り返し処理の占める割合が非常に多く、繰り返し処理のコードサイズの削減は有効である。プログラムをROM化する場合にはプログラムのコード効率がチップサイズに直接影響を及ぼすので、この実施の形態1による情報処理装置はコスト削減にも大きく寄与する。また、プログラムを簡素化すること



ができるので、プログラム開発の生産性を向上できるとともにバグの混入を低減できるという効果もある。

#### 【0135】

この実施の形態1による情報処理装置は、基本的に、リピートブロックを所定の繰り返し回数処理するハードウェアにより実現されたリピート機能と、リピートブロックの途中のリピート処理終了命令でリピート処理を終了するとともに、リピートブロックの次の命令に命令処理シーケンスを切り替える機能とを備えていればよく、上記した構成に限定されるものではない。どのようなハードウェア構成を有していても同様の効果を奏することができる。

#### 【0136】

また、この実施の形態1は、上記のものとは異なるパイプライン処理構成のものに適用しても良いし、VLIWプロセッサではなく通常のプロセッサやDSPに適用しても良い。

#### 【0137】

さらに、REP命令やSETRP命令のような、該当する命令の次の命令からリピート処理を開始する命令を実行せずに、指定されたアドレスからリピート処理を開始するように事前に設定しておくようにしても良い。

#### 【0138】

上述のREP命令では、リピートブロックの命令数が2の $n$ 乗( $n$ は整数)の場合に、レジスタ番号Rsrc1, Rsrc2に設定する設定値の計算が単純になるように、レジスタ番号Rsrc2に“リピートブロックの繰り返し回数-1”を、レジスタ番号Rsrc1に“リピート処理の最終サイクルで実行するリピートブロック内の命令数-1”を設定しているが、これに限定されるものではない。例えば、レジスタ番号Rsrc2に“リピートブロックの繰り返し回数”を設定しても良いし、レジスタ番号Rsrc1に“リピート処理の最終サイクルで実行するリピートブロック内の命令数”を設定してもよい。すなわち、ハードウェア的に計算でき対応関係が明確であれば、いかなるリピート命令も実現可能である。

#### 【0139】

リピート処理用に、フェッチした命令のアドレスと比較するための2つの比較器187, 172が独立して設けられているが、1つの比較器を設け、TRPT\_Cレジスタ175に保持されている値が非ゼロである場合には、比較器にフェッチした命令のアドレスをRPT\_Eレジスタ186に保持されている値とを比較させ、TRPT\_Cレジスタ175に保持されている値がゼロの場合には、比較器にフェッチした命令のアドレスをRPT\_Bレジスタ171に保持されている値とを比較させるように切り替えるセレクトを設けてもよい。

#### 【0140】

リピートブロックの開始の直前でなく、2命令以上前にリピート命令を実行するディレイドリピートを行うようにすれば（実施の形態6にて詳細に述べる）、リピートブロックのサイズの制約やリピート最終命令アドレスの制約は限定される。この場合、リピートブロックに関する各種の設定値、その設定方法、その計算方法等を変更すればよい。例えば、リピート命令をリピートブロックの2命令前に実行するのであれば、レジスタ番号Rsrc1に“リピート処理の最終サイクルで実行するリピートブロック内の命令数”を設定して同様の計算を行えば、RPT\_Bは計算できる。RPT\_Eは、同様に、リピート命令のアドレスからの変位値を指定することにより計算できる。また、RPT\_Sは、ラッチ192の出力ではなくインクリメンタ193の出力を用いれば、ハードウェア量の増加なしに計算できる。このようなディレイドリピート制御を行えば、リピートブロックの命令のフェッチ開始とリピート命令実行によるリピート関連の制御レジスタ等の設定とのパイプライン処理による時間差が短くなる。従って、2命令前にリピート命令を実行する場合、この実施の形態1による情報処理装置では動作保証できなかった、リピートブロックが2つの32ビット命令から成る場合や、レジスタ番号Rsrc2により指定される値が“0”でレジスタ番号Rsrc1により指定される値が“2”の場合にも、正しく動作できるようになる。

#### 【0141】

また、この実施の形態1による情報処理装置は、複数レベルのリピート機能を備えていてもよい。

#### 【0142】

さらに、ダイナミックに変化し得る繰り返し回数が設定できるようにレジスタで繰り返し回数を指定するリピート命令の代わりに、レジスタ番号  $Rsrc1$ ,  $Rsrc2$  で設定する値に相当する値を即値データで設定するリピート命令を実装した場合、 $Rsrc1$ ,  $Rsrc2$  によるレジスタ設定が不要となり、さらにレジスタ設定等のオーバヘッドが削減でき、リピート命令の前処理を省略することができリピート処理を高速化できる。

## 【0143】

実施の形態 2.

この発明の実施の形態 2 による情報処理装置は、上記実施の形態 1 と異なり、リピート処理最終命令の処理後、リピートブロックの次の命令に命令処理シーケンスを切り替えずに、リピートブロックの最終命令まで命令の実行を抑止するものである。この実施の形態 2 による情報処理装置は、基本的には、上記実施の形態 1 による情報処理装置とほぼ同一の構成を有しており、以下では、その説明を省略し異なる部分についてのみ説明する。

## 【0144】

図 20 はこの発明の実施の形態 2 による情報処理装置が用いる PSW の構成を示す図である。この実施の形態 2 による PSW は RE ビット 601 を備えている点で上記実施の形態 1 によるものと相違する。RE ビット 601 は、“1” の場合、リピート処理最終命令の処理後であってリピートブロック最終命令までの命令が処理中であることを示している。すなわち、“1” に設定された RE ビット 601 は、リピート処理が完了しており、命令の実行を抑止する命令の処理中であることを示している。実行開始時に RE ビット 601 が “1” である場合には、E ステージ 403 では、命令の実行は行われな（ノーオペレーション）。

## 【0145】

図 21 はこの実施の形態 2 による情報処理装置の PC 部 118 の構成を示すブロック図である。この実施の形態 2 による情報処理装置の PC 部 118 は、インクリメンタ 173、ラッチ 174 を有していない点が、上記実施の形態 1 による PC 部 118 と異なる。また、この実施の形態 2 による PC 部 118 は、ラッチ 174 から NPC 611 へ至る転送経路も有していない。

## 【 0 1 4 6 】

次に動作について説明する。

以下では図 1 6 に示したプログラム例を実行する場合の動作について説明する。図 2 2 は図 1 6 に示したプログラム実行時のパイプライン処理を示すタイミングチャート図であり、上記実施の形態 1 における図 1 8 に相当する。なお、クロックサイクル T 1 ～ T 9 に相当する部分は図 1 7 に示したものと同一である。この実施の形態 2 が上記実施の形態 1 と大きく異なる点は、リピート処理最終命令の処理後、I F ステージ 4 0 1 でリピートブロックの次の命令に命令処理シーケンスを切り替えずにリピートブロックの最終命令まで命令の実行を抑止することである。

## 【 0 1 4 7 】

T R E ビットは、制御部 1 1 2 内のレジスタ等（図示せず）に保持されており、P S W の R E ビット 6 0 1 に相当し、I F ステージ 4 0 1 で更新される。制御部 1 1 2 に設けられた P S W 部 2 2 1 の P S W ラッチ 2 2 2 の R P ビット 4 3 が “ 1 ”（リピート処理中）であり且つ P C 部 1 1 8 の T R P T \_ C レジスタ 1 7 5 に保持されている値が非ゼロである場合の処理は上記実施の形態 1 と同じである。リピート処理中に T R P T \_ C レジスタ 1 7 5 に保持されている値がゼロである場合、命令フェッチ部 1 0 2 が I 7 の命令をフェッチすると、P C 部 1 1 8 の比較器 1 7 2 は比較結果が一致した旨を示す一致信号を制御部 1 1 2 へ送出する。図 2 2 のクロックサイクル T 1 2 はこの状態を示している。制御部 1 1 2 は、比較器 1 7 2 から一致信号を受信すると T R E ビットを “ 1 ” にセットする。その後、命令フェッチ部 1 0 2 が I 9 の命令をフェッチする場合、比較器 1 8 7 は比較結果が一致した旨を示す一致信号を制御部 1 1 2 へ送出する。図 2 2 のクロックサイクル T 1 4 はこの状態を示している。制御部 1 1 2 は、比較器 1 8 7 から一致信号を受信すると T R E ビットをゼロクリアする。リピートブロックの最終命令が、リピート処理最終命令である場合すなわち R P T \_ E レジスタ 1 8 6 に保持されている値と R P T \_ B レジスタ 1 7 1 に保持されている値とが同一の場合、2 つの比較器 1 8 7, 1 7 2 は比較結果が一致した旨を示す一致信号を同時に制御部 1 1 2 へ送出する。この場合、制御部 1 1 2 は T R E ビットを “ 1

” にセットしない。T R E ビットの値は、命令実行抑止情報として対応する命令コードに従ってパイプライン転送される。リピート処理最終命令の次の命令からリピートブロックの最終命令までの処理の間、T R E ビットの値は“1”のまま保持される。

#### 【0148】

D ステージ 4 0 2 では、命令実行抑止情報が 1 の場合、情報処理装置は命令コードに依存せずノーオペレーションの実行制御情報を生成し、E ステージ 4 0 3 に発行する。

#### 【0149】

E ステージ 4 0 3 では、情報処理装置はリピート処理最終命令の次の命令からリピートブロックの最終命令まで命令の実行は行わない。リピート処理最終命令情報およびリピートブロック最終命令情報は、上記実施の形態 1 と同様、パイプライン転送される。リピート処理最終命令実行時、リピート処理最終命令がリピートブロック最終命令でないならば、制御部 1 1 2 は P S W ラッチ 2 2 2 に保持されている P S W の R E ビット 6 0 1 を“1”に設定する。R E ビット 6 0 1 が“1”の状態ではリピートブロック最終命令が実行されると、制御部 1 1 2 は P S W の R E ビット 6 0 1 をゼロクリアする。リピート処理最終命令がリピートブロック最終命令である場合には、制御部 1 1 2 は P S W の R E ビット 6 0 1 をセットすることなくリピート処理を終了する。また、P C 部 1 1 8 は、上記実施の形態 1 と同様に、R P T \_ C レジスタ 1 8 8 の内容を更新する。

#### 【0150】

リピート処理最終命令処理開始時、ならびに、P S W の R E ビット 6 0 1 が“1”の状態ではリピートブロック最終命令の処理開始時、P C 部 1 1 8 の N P C 6 1 1 の内容は、シーケンシャルに命令を実行する場合と同様に、インクリメント 1 9 3 でインクリメントされたアドレスに更新される。R E ビット 6 0 1 が“1”の状態ではリピートブロック最終命令の処理開始時には、上記実施の形態 1 と同様、P C 部 1 1 8 はラッチ 1 8 5 に保持されている値を N P C 6 1 1 に書き込む。

#### 【0151】

E ステージジャンプ時には、T R E ビットの値は、E ステージで管理している P S W の R E ビット 6 0 1 の情報に基づき更新され、T R E ビットの初期設定が行われる。

#### 【0 1 5 2】

上記したように、この実施の形態 2 による情報処理装置は、D ステージ 4 0 2 で実行抑止対象の命令のノーオペレーション化を行うが、実行の抑止は、D ステージ 4 0 2 に限定されず E ステージ 4 0 3 以前のどのパイプラインステージで行っても良い。ユーザから見えるハードウェアリソース（汎用レジスタ、アキュムレータ、制御レジスタやメモリの値）の更新さえ行われなければ、どのパイプラインステージで実行抑止対象の命令のノーオペレーション化が行われても良い。また、D ステージ 4 0 2 で実行抑止対象の命令のノーオペレーション化を行う代わりに、命令のフェッチ自体を抑止しても良いし、命令フェッチを行いフェッチした命令を特定の命令コードに置き換えてもよい。命令フェッチアドレスを含め P C 値が正しく管理されるように制御されさえすればよい。例えば、フェッチした命令を N O P 命令の並列実行コードに置き換えれば、特別なノーオペレーション化の処理は必要なくなる。

#### 【0 1 5 3】

また、E ステージ 4 0 3 において実行抑止対象の命令のノーオペレーション化を行っても良い。E ステージ 4 0 3 で実行抑止判定を行う場合には、命令実行抑止情報のパイプライン転送は必須ではない。ただし、シーケンシャルに実行する 2 命令等の命令デコードに複数クロックサイクルを要する場合もあるので、無駄なクロックサイクルが増える可能性がある。

#### 【0 1 5 4】

R E ビット 6 0 1 は、実行抑止中に外部割り込み等を受け付ける場合にも正しく処理が復帰できるように、P S W に割り付けられている。従って、リピート処理最終命令実行後からリピートブロック最終命令の実行後までの間割り込みを受け付けられないような制御を行えば、R E ビット 6 0 1 は不要である。

#### 【0 1 5 5】

以上のように、この実施の形態 2 による情報処理装置は、上記実施の形態 1 と

異なり、リピート処理最終命令の処理後、リピートブロックの次の命令に命令処理シーケンスを切り替えずに、リピートブロックの最終命令まで命令の実行を抑制するものであるが、上記実施の形態 1 によるものと同様な効果を奏する。すなわち、繰り返し回数がダイナミックに変化する場合の繰り返し処理において、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる。従って、プログラムを ROM 化する場合のコストも削減できるうえに、プログラム開発の生産性を向上することができる。

#### 【0156】

さらに、この実施の形態 2 による情報処理装置の PC 部 118 は、図 10 に示すインクリメンタ 173 やラッチ 174 を備えていないので、上記実施の形態 1 に比べさらに少ないコストで前述の効果をを得ることができる。ただし、この実施の形態 2 による情報処理装置は、実行抑止対象の命令に関してもパイプライン処理を行うので、上記実施の形態 1 の場合に比べて多少無駄なクロックサイクルが生じる。

#### 【0157】

実施の形態 3.

この発明の実施の形態 3 による情報処理装置は、上記実施の形態 1 と異なり、リピート処理最終命令の処理後、IF ステージ 401 ではなく E ステージ 403 でリピートブロックの次の命令に命令処理シーケンスを切り替えるものである。この実施の形態 3 による情報処理装置は、基本的には、上記実施の形態 1 による情報処理装置とほぼ同一の構成を有しており、以下では、その説明を省略し異なる部分についてのみ説明する。

#### 【0158】

なお、この実施の形態 3 による情報処理装置の PC 処理部 118 は図 10 に示すものと同一の構成を有する。

#### 【0159】

次に動作について説明する。

以下では図 16 に示したプログラム例を実行する場合の動作について説明する

。図23は図16に示したプログラム実行時のパイプライン処理を示すタイミングチャート図であり、上記実施の形態1における図18に相当する。なお、クロックサイクルT1～T9に相当する部分は図17に示したものと同一である。この実施の形態3が上記実施の形態1と異なる点は、リピート処理最終命令の処理後のリピートブロックの次の命令への命令処理シーケンスの切り替えを、IFステージ401ではなくEステージ403で行うことにある。

#### 【0160】

REP命令の処理およびリピート中の処理は、IFステージ401におけるI7の命令のフェッチ後I10の命令のフェッチを開始せず、シーケンシャルな命令フェッチを継続することを出いて、上記実施の形態1とまったく同様に行われる。また、この実施の形態3による情報処理装置は、上記実施の形態1と同じように、NPC191の更新を制御する。さらに、この実施の形態3による情報処理装置は、I7の命令実行開始時に上記実施の形態1と同一の制御を行う。情報処理装置は、また、サブルーチンコール時の戻り先アドレスとして、NPC191に保持されている値を参照する。従って、実施の形態3による情報処理装置は、上記実施の形態1による情報処理装置と同様に、I7の命令がサブルーチンコールを行う命令を含む場合でも、I10の命令のアドレスを戻り先アドレスとして参照することができる。

#### 【0161】

この実施の形態3の情報処理装置は、I7の命令のEステージ403での実行時（図23のクロックサイクルT14）に、リピートブロックの次の命令であるI10の命令へのジャンプ処理を行う。実行される命令で特定されるオペレーションとは独立して、制御部112は、リピート処理最終命令情報に基づき、ハードウェア的にジャンプ処理のための制御信号を生成する。クロックサイクルT14において、PC部118は、ラッチ174に保持されているI10の命令のアドレスをJAバス323を介してIAレジスタ181およびNPC191へ転送し、ジャンプ処理を行う。リピート処理最終命令の実行に複数クロックサイクルを要する場合には、情報処理装置はその最終クロックサイクルでジャンプ処理を行う。



## 【 0 1 6 2 】

ただし、リピート処理最終命令がリピートブロックの最終命令である場合には、この実施の形態 3 による情報処理装置は単にリピートブロックの次の命令を実行すれば良い。無駄なオーバーヘッド発生を抑止するため、情報処理装置はこの場合には E ステージ 4 0 3 でのジャンプ処理を行わない。なお、この実施の形態 3 による情報処理装置は、上記実施の形態 1 とは異なり、T R P T \_ C レジスタ 1 7 5 の内容がゼロであっても命令フェッチアドレスが R P T \_ E レジスタ 1 8 6 の内容と一致した場合、リピートブロック最終命令情報として一致した情報を出力するように制御する。情報処理装置は、実行する命令がリピート処理最終命令であることを示すリピート処理最終命令情報と、実行する命令がリピートブロック最終命令であることを示すリピートブロック最終命令情報とに基づき、リピート処理最終命令がリピートブロックの最終命令であるか否かを判定し、そうである場合には、単にリピートブロックの次の命令を実行する。

## 【 0 1 6 3 】

リピート処理最終命令が E ステージ 4 0 3 でジャンプ処理を行う命令である場合には、情報処理装置は、命令実行に伴うジャンプ処理を優先し、リピート処理に関するジャンプ処理は行わない。

## 【 0 1 6 4 】

以上のように、この発明の実施の形態 3 による情報処理装置は、I F ステージ 4 0 1 ではなく E ステージ 4 0 3 でリピート処理最終命令実行後の命令処理シーケンスの切り替え（リピートブロックの次の命令へのジャンプ）を行うが、上記実施の形態 1 と同じ効果を奏することができる。すなわち、繰り返し回数がダイナミックに変化するとともにリピートブロックの途中でリピート処理を終了する場合において、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる。従って、プログラムを R O M 化する場合のコストも削減できるうえに、プログラム開発の生産性を向上することができる。

## 【 0 1 6 5 】

なお、リピート処理最終命令としてサブルーチンコール命令を用いることを禁

止すれば、ラッチ 1 7 4 から NPC 1 9 1 へ至る転送経路は不要である。

【 0 1 6 6 】

また、RPT\_E レジスタ 1 8 6 に保持されている値を 1 インクリメントするために専用のインクリメンタ 1 7 3 を PC 部 1 1 8 に設けているが、インクリメンタ 1 7 3 の代わりに既存のリソースを使用してハードウェアリソースを削減することも可能である。この場合、性能およびハードウェアコストを考慮して、ハードウェア構成や制御方法を決定することが好ましい。ただし、そのハードウェアリソースを実行時に使用する命令は、リピート処理最終命令として実行を禁止する必要がある。

【 0 1 6 7 】

なお、図 2 3 に示す例では、IF ステージ 4 0 1 において I 8 以降の命令のフェッチは継続して行われているが、これらは無駄なフェッチであるので、I 8 以降の命令フェッチを抑止するように制御しても良い。

【 0 1 6 8 】

実施の形態 4 .

この発明の実施の形態 4 による情報処理装置は、上記実施の形態 3 と同様に、リピート処理最終命令の処理後、リピートブロックの次の命令に命令処理シーケンスを IF ステージ 4 0 1 ではなく E ステージ 4 0 3 で切り替えるが、リピートに関するジャンプ処理をリピート処理最終命令の実行時に行うのではなく、実行後の次のクロックサイクルで行うものである。この実施の形態 4 による情報処理装置は、基本的には、上記実施の形態 1 による情報処理装置と同一の構成を有しており、以下では、その説明を省略し異なる部分についてのみ説明する。

【 0 1 6 9 】

なお、この実施の形態 4 による情報処理装置の PC 処理部 1 1 8 は図 2 1 に示すものと同一の構成を有する。

【 0 1 7 0 】

次に動作について説明する。

以下では図 1 6 に示したプログラム例を実行する場合の動作について説明する。図 2 4 は図 1 6 に示したプログラム実行時のパイプライン処理を示すタイミン

グチャート図であり、上記実施の形態 1 における図 1 8 に相当する。なお、クロックサイクル T 1 ~ T 9 に相当する部分は図 1 7 に示したものと同一である。上記したように、この実施の形態 4 による情報処理装置は、上記実施の形態 3 と同様に、リピート処理最終命令処理後のリピートブロックの次の命令へのシーケンス切り替えを E ステージ 4 0 3 で行う。上記実施の形態 3 と異なる点は、リピートに関するジャンプ処理をリピート処理最終命令の実行時に行うのではなく、実行後の次のクロックサイクルで行うことにある。

#### 【 0 1 7 1 】

図 2 4 のクロックサイクル T 1 4 で I 7 のリピート処理最終命令が実行される。図 2 4 において、“( j m p )” はリピートブロックの次の命令へのジャンプ処理を示している。クロックサイクル T 1 4 において、制御部 1 1 2 は、実行される命令で特定されるオペレーションとは独立して、リピート処理最終命令情報に基づき、ハードウェア的にリピートブロックの次の命令へのジャンプ処理を行うための制御信号を生成し、クロックサイクル T 1 5 において、情報処理装置はジャンプ処理を行う。クロックサイクル T 1 5 において、PC 部 1 1 8 は R P T \_ E レジスタ 1 8 6 に保持されている値を S 3 バス 3 0 3 へ出力し、第 1 演算部 1 1 6 の A A ラッチ 1 5 1 は S 3 バス 3 0 3 上に出力されたその値を取り込む。また、第 1 演算部 1 1 6 の A B ラッチ 1 5 2 には 1 がセットされる。A L U 1 5 3 は A A ラッチ 1 5 1 および A B ラッチ 1 5 2 に保持されている値からリピートブロックの次の命令のアドレスを生成し、このアドレスを J A バス 3 2 3 へ出力し、PC 部 1 1 8 は J A バス 3 2 3 上に出力されたそのアドレスを I A レジスタ 1 8 1 および N P C 6 1 1 に取り込む。この結果、情報処理装置はジャンプ処理を行う。

#### 【 0 1 7 2 】

ただし、リピート処理最終命令がリピートブロックの最終命令である場合には、この実施の形態 4 による情報処理装置は単にリピートブロックの次の命令を実行するだけで良い。無駄なオーバーヘッド発生を抑止するため、情報処理装置は、この場合、E ステージ 4 0 3 でのジャンプ処理を行わない。なお、この実施の形態 4 による情報処理装置は、上記実施の形態 3 と同様に、T R P T \_ C レジス

タ 1 7 5 の内容がゼロであっても命令フェッチアドレスが R P T \_ E レジスタ 1 8 6 の内容と一致した場合、リピートブロック最終命令情報として一致した情報を出力するように制御する。情報処理装置は、実行する命令がリピート処理最終命令であることを示すリピート処理最終命令情報と、実行する命令がリピートブロック最終命令であることを示すリピートブロック最終命令情報とに基づき、リピート処理最終命令がリピートブロックの最終命令であるか否かを判定することができる。

## 【 0 1 7 3 】

リピート処理最終命令が E ステージ 4 0 3 でジャンプ処理を行う命令である場合、この実施の形態 4 による情報処理装置は、命令実行に伴うジャンプ処理を優先しリピート処理に関するジャンプ処理を行わない。

## 【 0 1 7 4 】

上記したように、この実施の形態 4 による情報処理装置の P C 部 1 1 8 は図 2 1 に示す構成を有しており、図 1 0 に示すインクリメンタ 1 7 3 を備えていない。従って、この実施の形態 4 による情報処理装置は、リピート処理最終命令としてサブルーチンコール命令を実行することはできない。図 2 4 において、I 7 のリピート処理最終命令実行時の N P C 値は、I 1 0 ではなく I 8 の命令のアドレスとなっている。それ故、リピート処理最終命令としてサブルーチンコール命令を実行できるようにする場合には、P C 部 1 1 8 にインクリメンタを追加すればよい。

## 【 0 1 7 5 】

以上のように、この発明の実施の形態 4 による情報処理装置は、I F ステージ 4 0 1 ではなく E ステージ 4 0 3 でリピート処理最終命令実行後の命令処理シーケンスの切り替え（リピートブロックの次の命令へのジャンプ）を行い、さらに、リピートに関するジャンプ処理をリピート処理最終命令の実行時に行うのではなく、実行後の次のクロックサイクルで行うものであるが、上記実施の形態 1 と同じ効果を奏することができる。すなわち、繰り返し回数がダイナミックに変化するとともにリピートブロックの途中でリピート処理を終了する場合において、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向

上をはかることができるとともに、プログラムサイズを大幅に削減できる。従って、プログラムをROM化する場合のコストも削減できるうえに、プログラム開発の生産性を向上することができる。

#### 【0176】

実施の形態5.

この発明の実施の形態5による情報処理装置は、上記実施の形態1～4に示したようなリピートブロックの最終処理サイクルでのリピート処理終了判定をアドレスに基づいて行うのではなく、リピートブロックの最終処理サイクルでの処理終了判定をリピートブロックで実行される命令数（以下実行命令数と称する）に基づいて行うものである。この実施の形態5による情報処理装置は、基本的には、上記実施の形態1による情報処理装置と同一の構成を有しており、以下では、その説明を省略し異なる部分についてのみ説明する。

#### 【0177】

図25はこの実施の形態5による情報処理装置のリピート関連の制御レジスタを示す図である。図25と図1との比較からわかるように、この実施の形態5による制御レジスタCR6 701はRPT\_Bの代わりに“リピートブロックの最終リピートサイクルで実行する命令数-1”に等しい初期値に設定されるRPT\_BCを保持している。すなわち、この実施の形態5による情報処理装置は、アドレスではなく実行命令数に基づいてリピート処理の終了を判定する。

#### 【0178】

図26はこの実施の形態5による情報処理装置のPC部118の構成を示すブロック図である。図に示すように、PC部118は、上記実施の形態1によるものとは異なり、図10に示すRPT\_Bレジスタ171関連のハードウェアを備えていない。図26において、711はRPT\_BCを保持するRPT\_BCレジスタであり、713はラッチ712を介して入力されたRPT\_BCレジスタ711に保持されている値を1デクリメントするデクリメンタ（DEC）であり、714はIFステージ401でRPT\_BCを先行して更新するためにRPT\_BCを保持するTRPT\_BCレジスタであり、716はラッチ715を介して入力されたTRPT\_BCレジスタ714に保持されている値を1デクリメン

トするデクリメンタ (DEC) である。PC部118は、1命令毎にRPT\_BCを更新するため、IFステージ401でのRPT\_BC更新用とEステージ403でのRPT\_BC更新用に独立した2つのデクリメンタ713, 716を備えている。また、717はTRPT\_BCレジスタ714に保持されている値がゼロになったことを検出するゼロ検出回路 (ZERO) である。PC部118は、また、ジャンプが起こった際にパイプラインの前段での前処理をキャンセルして初期化するために、ラッチ718を介してRPT\_BCレジスタ711からTRPT\_BCレジスタ714へ至る転送経路を備えている。なお、カウント手段は、TRPT\_BCレジスタ714、ラッチ715およびデクリメンタ716からなる。また、第1カウント手段は、TRPT\_Cレジスタ175、ラッチ176およびデクリメンタ177からなり、第2カウント手段は、TRPT\_BCレジスタ714、ラッチ715およびデクリメンタ716からなる。

#### 【0179】

また、PC部118は、次に実行する命令のPC値を保持するラッチ192からJAバス323へ至る出力経路と、JAバス323からRPT\_Sレジスタ184へ至る転送経路とを備えている。

#### 【0180】

次に動作について説明する。

図27および図28は、図16に示したプログラム実行時のパイプライン処理を示すタイミングチャート図であり、上記実施の形態1の図17および図18に相当する。上記したように、この実施の形態5による情報処理装置は、上記実施の形態1～4と異なり、実行命令数に基づきリピートブロックの最終処理サイクルでのリピート処理終了判定を行う。

#### 【0181】

この実施の形態5による情報処理装置は、上記実施の形態1と同様に、図16に示すI5のREP命令を2クロックサイクルかけて実行する。情報処理装置は、第1クロックサイクルにおいて、図25に示すRPT\_EおよびRPT\_Cを設定し、第2クロックサイクルにおいて、図25に示すRPT\_BC、RPT\_Sおよび図2に示すPSWのRPビット43を設定する (物理的なレジスタでな

く論理的な制御レジスタを示している)。

#### 【0182】

情報処理装置は、上記実施の形態1と同様に、第1クロックサイクルの処理を行う。第2クロックサイクルでは、I5のREP命令のレジスタ番号Rsrc1で指定される汎用レジスタR11 12に保持されている値が、S2バス302、ラッチ169、セクタ155、D1バス311を介して、RPT\_BCレジスタ711およびTRPT\_BCレジスタ714へ転送され、さらにその後、RPT\_BCレジスタ711の値がラッチ718へ転送される。上記処理と並行して、PC部118は、リピートブロックの先頭の命令のアドレスとなるREP命令の次の命令のアドレスを、NPC191からラッチ192およびJAバス323を介してRPT\_Sレジスタ184へ転送し、さらにその後、ラッチ185へ転送する。また、制御部112は、PSWラッチ222のRPビット43を“1”に設定する。

#### 【0183】

情報処理装置は、TRPT\_Cレジスタ175に保持されている値がゼロになるまで、上記実施の形態1とまったく同じ処理を行う。TRPT\_Cレジスタ175に保持されている値がゼロになると、その後、命令フェッチ部102が1命令フェッチする毎に、デクリメンタ716はTRPT\_BCレジスタ714に保持されている値を1デクリメントする。そして、TRPT\_BCレジスタ714に保持されている値がゼロになると、制御部112は命令処理シーケンスの切り替えを行う。すなわち、制御部112は、I7の命令フェッチ後I10の命令のフェッチを行うようにPC部118を制御する。情報処理装置は、上記実施の形態1とまったく同じように、命令処理シーケンスの切り替えを行う。TRPT\_Cレジスタ175に保持されている値がゼロの状態ではフェッチされた命令に関しては、リピートブロックの最終処理サイクルであることを示している情報が命令のパイプライン処理に同期して転送される。Eステージ403では、リピートブロックの最終処理サイクル中、1命令の実行が完了する毎に、RPT\_BCレジスタ711に保持されている値のデクリメントが行われる。ただし、I7のリピート処理最終命令処理時にはそのデクリメントは行われない。制御部112は、

I 7 のリピート処理最終命令処理時に P S W ラッチ 2 2 2 の R P ビット 4 3 をゼロクリアする。

【 0 1 8 4 】

E ステージ 4 0 3 でジャンプが起こった際には、P C 部 1 1 8 は、R P T \_ B C レジスタ 7 1 1 に保持されている値をラッチ 7 1 8 を介して T R P T \_ B C レジスタ 7 1 4 へ転送する。

【 0 1 8 5 】

以上のように、この発明の実施の形態 5 による情報処理装置は、上記実施の形態 1 ～ 4 とは異なり、リピートブロックの最終処理サイクルでのリピート処理終了判定をアドレスではなく実行命令数に基づいて行うものであるが、上記実施の形態 1 と同じ効果を奏することができる。すなわち、繰り返し回数がダイナミックに変化するとともにリピートブロックの途中でリピート処理を終了する場合において、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる。従って、プログラムを R O M 化する場合のコストも削減できるうえに、プログラム開発の生産性を向上することができる。ただし、この実施の形態 5 による情報処理装置は、リピートブロックの最終処理サイクルでのリピート処理終了判定を実行命令数に基づいて行うので、最後にリピートブロックの先頭の命令のアドレスに戻ってからリピート処理終了命令を実行するまでの間（R P T \_ C レジスタ 1 8 8 に保持されている値がゼロの状態）に条件分岐命令がある等、リピートブロックの最終処理サイクルで実行される命令数がダイナミックに変化する場合には適用できない。従って、この実施の形態 5 による情報処理装置のこのような状況での使用は禁止する必要がある。

【 0 1 8 6 】

なお、この実施の形態 5 による情報処理装置は、リピートブロックの最終処理サイクルであるか否かを E ステージ 4 0 3 で判定してもよい。この場合、情報のパイプライン転送は不要である。

【 0 1 8 7 】

また、この実施の形態 5 による情報処理装置は、I F ステージ 4 0 1 での R P



T\_\_BC更新用とEステージ403でのRPT\_\_BC更新用に独立した2つのデクリメンタを備えている代わりに、IFステージ401でのRPT\_\_BC更新用に1つのデクリメンタを備え、TRPT\_\_BCレジスタ714に保持されている値をRPT\_\_BCレジスタ711へパイプライン転送してもよい。

#### 【0188】

さらに、この実施の形態5による情報処理装置は、RPT\_\_BCを所定の設定値からデクリメントする代わりに、ゼロからインクリメントするようにして所定の設定値との一致判定を行うような制御方法を用いてもよい。

#### 【0189】

実施の形態6.

この発明の実施の形態6による情報処理装置は、リピート処理中に実行される命令数（すなわち実行命令数）のみに基づきリピート処理終了判定を行うとともに、リピートブロックを処理するために、リピートブロックの開始直前にリピート命令を実行するのではなく、リピートブロックの2命令以上前に設けたディレイドリピート命令（遅延リピート）を実行するものである。この実施の形態6による情報処理装置は、基本的には、上記実施の形態1による情報処理装置と同一の構成を有しており、以下では、その説明を省略し異なる部分についてのみ説明する。

#### 【0190】

図29はこの実施の形態6による情報処理装置のリピート関連の制御レジスタを示す図である。図29と図1との比較からわかるように、この実施の形態6による情報処理装置は、制御レジスタCR6（RPT\_\_B）を有していない。制御レジスタCR7 801は、“リピート処理中に実行される命令数”に相当するRPT\_\_Cを保持する。この実行命令数は32ビット命令を1命令としてカウントされる。RPT\_\_Cは、リピート処理開始時に実行される総実行命令数に設定され、リピート処理中に1命令実行毎に1デクリメントされる。RPT\_\_Cがゼロになるとリピート処理は終了する。

#### 【0191】

図30はこの実施の形態6による情報処理装置のPC部118の構成を示すブ

ロック図である。図に示すように、PC部118は、上記実施の形態1によるものとは異なり、図10に示すRPT\_\_Bレジスタ171関連のハードウェアを有していない。さらに、RPT\_\_Cレジスタ関連のハードウェア構成は上記実施の形態1によるものとは大きく異なる。図30において、813はラッチ812を介して入力されたRPT\_\_Cレジスタ811に保持されている値を1デクリメントするデクリメンタ（DEC）であり、814はIFステージ401でカウント値RPT\_\_Cを先行して更新するためにRPT\_\_Cを保持するTRPT\_\_Cレジスタであり、816はラッチ815を介して入力されたTRPT\_\_Cレジスタ814に保持されている値を1デクリメントするデクリメンタ（DEC）である。PC部118は、1命令毎にRPT\_\_Cを更新するため、IFステージ401でのRPT\_\_C更新用とEステージ403でのRPT\_\_C更新用に独立した2つのデクリメンタ813、816を備えている。また、817はTRPT\_\_Cレジスタ814に保持されている値が1になったことを検出する1検出回路（ONE）である。また、ジャンプが起こった際にパイプラインの前段での前処理をキャンセルして初期化するために、PC部118は、ラッチ818を介してRPT\_\_Cレジスタ811からTRPT\_\_Cレジスタ814へ至る転送経路を備えている。また、RPT\_\_Cレジスタ811およびTRPT\_\_Cレジスタ814にD1バス311からの転送経路が設けられている。

#### 【0192】

次に動作について説明する。

図31および図32はこの実施の形態6による情報処理装置のディレイドリピート命令の2つの例“DREP Rsrc, disp16”および“DREPI imm8, disp16”の命令フォーマットをそれぞれ示す図である。これらの命令はロングフォーマットの命令である。これらの図において、831、836はFMビットであり、832、837はオペレーションコードであり、833は予約ビットであり、834（Rsrc）はリピートブロック中の実行命令数（実行される32ビット命令の数）を保持するレジスタを指定するレジスタ番号であり、838（imm8）はリピートブロック中の実行命令数を示す8ビットの即値データであり、835、839（disp16）はリピートブロック最終

命令のアドレスのリピート命令のアドレスからの変位値である。

#### 【0193】

ディレイドリピート命令は、リピート処理の開始に関して1命令遅延した制御を行う命令である。ディレイドリピート命令の次々命令からリピートブロックが始まる。ディレイドリピート命令の次の命令は、ディレイドリピート命令実行後必ず実行されるものとする。すなわち、外部割り込み要求がアサートされており、割り込み受け付け可能状態（図2に示すPSWのIEビット42が“1”の場合）であっても、情報処理装置は、割り込みの受け付けを抑止する。また、ディレイドリピート命令直後の命令は命令処理シーケンスを切り替えるジャンプ命令等であってはならないし、リピートブロックは2命令以上の命令を含んでいなければならない。しかしながら、リピートブロック内の実行命令数が1であっても、情報処理装置は正しくリピート処理を行える。

#### 【0194】

図33は図16に示したものと同じく上記式(1)に従って積和演算を行うアセンブラで記述されており、DREP命令を含むプログラム例を示す図である。汎用レジスタR10 11は積和演算回数Nを保持しており、汎用レジスタR8 9および汎用レジスタR9 10は積和対象の2つの配列の先頭A[0], B[0]のアドレスをそれぞれ保持している。I1a~I3の命令が前処理およびリピート命令であり、I4~I7の命令がリピートブロックである。リピートブロックの開始命令であるI4の2命令前でDREP命令が実行される。

#### 【0195】

図34および図35はこの実施の形態6によるリピート処理中のパイプライン処理を示すタイミングチャート図である。これら2つの図は連続した図であり、わかりやすいようにクロックサイクルT9およびT10は重複して示している。図の上部4行は各パイプラインステージの処理を示しており、残りの下部はリピート処理に関係するレジスタに保持されている値や信号の値を示している。また、簡単のため、各命令およびアドレスをI2等命令の仮称で示している。“IA==RPT\_\_E”は比較器187の比較結果を示しており、“TRPT\_\_C==1”は1検出回路817の出力を示す。また、図34および図35は10回の積

和を行う場合のパイプライン処理を示している。図示した例は、リピート処理最終命令が I 5 であり、I 4 ~ I 7 の 4 命令のリピーブロックの処理を 2 回さらに I 4, I 5 の 2 命令を実行したのち、I 8 の実行を行うものである。

#### 【0196】

図 34 に示すように、情報処理装置は 2 クロックサイクル T 3, T 4 で I 2 の D R E P 命令を実行する。また、情報処理装置は、クロックサイクル T 3 において、図 29 に示す R P T \_ E および R P T \_ C を設定し、P S W ラッチ 2 2 2 の R P ビット 4 3 を設定する。P C 部 1 1 8 は制御部 1 1 2 の制御のもとで E P C 1 9 4 に保持されている値を S 3 バス 3 0 3 へ出力し、第 1 演算部 1 1 6 の A A ラッチ 1 5 1 は S 3 バス 3 0 3 上に出力されたその値を取り込む。また、d i s p 1 6 8 3 5 は第 1 演算部 1 1 6 の A B ラッチ 1 5 2 へ転送される。第 1 演算部 1 1 6 の A L U 1 5 3 は、A A ラッチ 1 5 1 および A B ラッチ 1 5 2 に保持されている値同士を加算し、加算結果であるリピートブロックの最終命令のアドレスを J A バス 3 2 3 に出力する。P C 部 1 1 8 の R P T \_ E レジスタ 1 8 6 は J A バス 3 2 3 上に出力されたそのアドレスを取り込む。その後、インクリメンタ 1 7 3 は、R P T \_ E レジスタ 1 8 6 に保持されている値を 1 インクリメントし、インクリメント結果はラッチ 1 7 4 へ転送される。また、情報処理装置は、R s r c 8 3 4 で指定された汎用レジスタ R 1 0 1 1 に保持されている値を S 1 バス 3 0 1、ラッチ 1 6 9、セクタ 1 5 5 および D 1 バス 3 1 1 を介して R P T \_ C レジスタ 8 1 1 および T R P T \_ C レジスタ 8 1 4 へ転送する。その後、R P T \_ C レジスタ 8 1 1 の値はラッチ 8 1 8 へ転送される。そして、制御部 1 1 2 は、P S W ラッチ 2 2 2 の P R ビット 4 3 を “1” にセットする。

#### 【0197】

クロックサイクル T 4 において、制御部 1 1 2 は R P T \_ S レジスタ 1 8 4 の設定を行う。P C 部 1 1 8 は N P C 1 9 1 に保持されている値を S 3 バス 3 0 3 へ出力し、第 1 演算部 1 1 6 の A A ラッチ 1 5 1 は S 3 バス 3 0 3 上に出力されたその値を取り込む。一方、制御部 1 1 2 は、第 1 演算部 1 1 6 の A B ラッチ 1 5 2 にハードウェア的に “1” を設定する。第 1 演算部 1 1 6 の A L U 1 5 3 は、A A ラッチ 1 5 1 および A B ラッチ 1 5 2 に保持されている値同士を加算し、

加算結果であるリピートブロックの先頭の命令のアドレスをセクタ155、D1バス311を介してPC部118のRPT\_\_Sレジスタ184へ転送する。その後、RPT\_\_Sレジスタ184に保持されている値はラッチ185へ転送される。

#### 【0198】

この実施の形態6による情報処理装置は、I2のDREP命令のデコードを開始した段階でI3の命令フェッチを開始するが、DREP命令のDステージでの処理が終了するまで、I4の命令フェッチの開始を抑止する。DREP命令のデコードがクロックサイクルT3で終了するので、情報処理装置はI4の命令フェッチをクロックサイクルT4で行う。DREP命令の実行により、クロックサイクルT3でRPT\_\_E、TRPT\_\_C、PSWラッチ222のRPビット43の設定が行われる。従って、この実施の形態6による情報処理装置は、リピートブロックの先頭命令であるI4の命令フェッチからディレイドリピート命令で設定された内容でリピート処理を行うことが可能である。

#### 【0199】

上記した処理例では、リピートブロック内の10個の命令が実行される。リピート処理中（PSWラッチ222のRPビット43が“1”の場合）は、命令フェッチ部102が32ビットの命令コードをフェッチする度に、PC部118のデクリメンタ816は、ラッチ815を介して受け取ったTRPT\_\_Cレジスタ814に保持されている値を1デクリメントし、デクリメント結果はTRPT\_\_Cレジスタ814に書き戻される。そして、1検出回路817は、TRPT\_\_Cレジスタ814に保持されている値が“1”になったか否かを判定する。リピート処理中にリピートブロックの命令が実行されると、デクリメンタ813は、各32ビット命令の処理の最終クロックサイクルにおいてRPT\_\_Cレジスタ811に保持されている値を1デクリメントし、デクリメント結果はTRPT\_\_Cレジスタ814に書き戻される（なお、I3の命令の処理時はデクリメントしない）。その後、RPT\_\_Cレジスタ811の値はラッチ818へ転送される。TRPT\_\_Cレジスタ814に保持されている値が“1”になるまで、この実施の形態6による情報処理装置は、上記実施の形態1と同様に、リピートブロックのR

ピート処理を行う。この実施の形態6によるリピートブロックのリピート処理方法は上記実施の形態1によるものと同じである。また、RPT\_Eレジスタ186に保持されている値とIAレジスタ181に保持されている値が一致した場合、PC部118は、次にフェッチする命令のアドレスとしてラッチ185に保持されている値をJAバス323を介してIAレジスタ181へ転送し、命令フェッチシーケンスの切り替えを行う。

#### 【0200】

TRPT\_Cレジスタ814に保持されている値が“1”の状態ではフェッチしている命令は、リピートブロック内で最後に実行される命令すなわちリピート処理最終命令である。従って、この場合、情報処理装置は次にフェッチする命令をリピートブロックの次の命令に切り替える。この実施の形態6による情報処理装置は、上記実施の形態1と同様に、命令処理シーケンスの切り替えを行う。PC部118は、制御部112の制御のもとでラッチ174に保持されている値をJAバス323を介してIAレジスタ181へ転送する。TRPT\_Cレジスタ814に保持されている値は、リピート処理最終命令のフェッチ時にも、デクリメンタ816により1デクリメントされる。また、リピート処理最終命令の実行時には、制御部112はPSWラッチ222のRPビット43をゼロクリアする。

#### 【0201】

リピート処理最終命令がリピートブロックの開始命令やリピートブロックの最終命令であっても、情報処理装置は正しくリピート処理を行える。

#### 【0202】

また、リピートブロック最終命令処理情報およびリピート処理最終命令処理情報は、上記実施の形態1と同様に、対応する命令のパイプライン処理に同期して転送され、NPC191の更新やPSWラッチ222のRPビット43のゼロクリア処理に使用される。DREP命令直後のI3の命令を処理する際には、情報処理装置はRPT\_CがデクリメントされないようにEステージ403でその命令の処理を制御する。

#### 【0203】

Eステージ403でジャンプが起こった場合、PC部118は、制御部112

の制御のもとでRPT\_Cレジスタ811に保持されている値をラッチ818を介してTRPT\_Cレジスタ814へ転送する。従って、リピートブロック内でジャンプを禁止する場合には、この転送経路は不要である。

#### 【0204】

上述のように、この実施の形態6による情報処理装置は、リピートブロックで実行される命令数を管理することにより、リピートブロックの処理終了判定を行っている。従って、リピート処理途中で条件分岐を行う場合など、実行命令数がデータに依存する場合にはこの実施の形態6による情報処理装置は適用できないため、このような場合のリピート処理を禁止する必要がある。

#### 【0205】

なお、図32に示すDREPI命令を実行する時には、制御部112は、第1クロックサイクルにおいて、図9に示すPSW部221からD1バス311へ至る転送経路を使用して、図32のimm8 838で指定されるリピートブロック内の実行命令数をPC部118のRPT\_Cレジスタ811およびTRPT\_Cレジスタ814へ転送する。それ以外の処理は、上記したDREP命令の実行の場合とまったく同じである。

#### 【0206】

この実施の形態6には多くの変形例があり得る。

この実施の形態6では、DREP命令による設定が終了するまでI4の命令のフェッチを開始しないように制御している。フェッチした命令をプリデコードし、DREP命令、DREPI命令フェッチ後は、PSWラッチ222のRPビット43がセットされるまで、リピートブロックの開始命令のフェッチを抑止するように制御することにより、実施の形態6とは異なる命令キューのサイズや命令フェッチの制御方法を使用した場合にもこの実施の形態6を適用することが可能である。また、ディレイドリピート命令のプリデコードがタイミング的に厳しいような場合には、2命令分遅延するようなディレイドリピート命令を実装しても良い。

#### 【0207】

また、RPT\_C, RPT\_S, RPT\_E等の各種設定値の設定方法は、必

要な値を設定するどのようなものであっても良い。例えば、RPT\_\_Sレジスタ184へRPT\_\_Sを設定する場合、インクリメンタ193の出力をD1バス3111を介してRPT\_\_Sレジスタ184に設定するようにしても良い。

#### 【0208】

この実施の形態6による情報処理装置は、リピートブロックは2命令以上の場合を対象としている。これは、RPT\_\_Sレジスタ184の設定後RPT\_\_Sがラッチ185へ転送されるが、I4の命令のフェッチにおいて比較器187で一致が検出された場合にはラッチ185への設定がまだ完了していないためである。DREP命令実行直後にI4の命令のフェッチにおいて比較器187が一致を検出した場合には、例えばRPT\_\_Sレジスタ184に設定する値をJAバス323を介して直接IAレジスタ181へも転送するようにすれば、この実施の形態6による情報処理装置は、リピートブロックが1命令の場合にも対処可能である。

#### 【0209】

また、この実施の形態6による情報処理装置は、DREP命令直後には割り込みを受け付けない。しかし、図2に示すPSW等にDREP命令の直後の命令であることを示す情報を保持することにより、DREP命令直後に割り込みを受け付けた場合には、この情報に基づき、DREP命令の次の1命令（図33のI3）については、TRPT\_\_Cレジスタ814およびRTP\_\_Cレジスタ811に保持されている値のデクリメントを抑止するような制御を行うことにより割り込みの禁止制約を解除することができる。

#### 【0210】

RPT\_\_Cレジスタ811およびTRPT\_\_Cレジスタ814に“リピートブロックで実行される命令数”に等しいRPT\_\_Cを設定するようにしているが、代わりに、“リピートブロックで実行される命令数-1”を設定し、RPT\_\_Cがゼロであるか否かを判定するゼロ検出回路を1検出回路817の代わりに設けても良い。また、これに代わって、RPT\_\_Cをゼロからインクリメントして、設定した命令数との一致判定を行うようにしても良い。

#### 【0211】



この実施の形態6による情報処理装置は、ハードウェア制約により、DREP命令等を2クロックサイクルで実行するものであるが、ハードウェアを追加することにより、1クロックサイクルで実行することができる。また、これに代わって、ハードウェアを削減することにより、3クロックサイクル以上かけてDREP命令等を実行するように制御することも可能である。また、専用のリピート命令すなわちDREP命令等ですべての状態設定を行う必要もなく、プリミティブとなる命令で設定を行うようにしてもよい。

#### 【0212】

IFステージ401とEステージ403で独立してTRPT\_Cレジスタ814およびRPT\_Cレジスタ811に保持されている値（カウンタ値）を更新するために、独立したデクリメンタ816、813を備えているが、パイプライン処理に同期してカウンタ値をIFステージ401からEステージ403へ転送するようにすれば、デクリメンタ813は不要である。ただし、そのための転送経路の追加が必要となる。

#### 【0213】

また、この実施の形態6による情報処理装置はディレイドリピート命令を実装した例であるが、RPT\_Cに“リピートブロックで実行される命令数-1”を設定し、リピートブロックの処理の1命令目はRPT\_Cをデクリメントしない仕様のリピート命令（ディレイドリピート命令ではない）を実装すれば、実施の形態6による情報処理装置とほぼ同じハードウェア構成で、遅延制御を行うことなく途切れることのない命令実行を実現できる。ただし、リピート命令によるリピート関連のレジスタ等の設定前に1命令のフェッチが終了しているので、1命令でリピートブロックを構成したり、1命令のみの実行でリピート処理を終了することはできない。

#### 【0214】

以上のように、この実施の形態6による情報処理装置は、リピートブロックで実行される命令数を管理することにより、リピートブロックの処理終了判定を行うとともに、リピートブロックを処理するために、リピートブロックの開始直前にリピート命令を実行するのではなく、リピートブロックの2命令以上前に設け

たディレイドリピート命令（遅延リピート）を実行するものであるが、上記実施の形態 1 による情報処理装置と同様の効果が得られる。すなわち、繰り返し回数がダイナミックに変化するとともにリピートブロックの途中でリピート処理を終了する場合において、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる。従って、プログラムを ROM 化する場合のコストも削減できるうえに、プログラム開発の生産性を向上することができる。また、リピートブロックの実行命令数を静的に設定できる場合すなわち D R E P I 命令を使用可能な場合でも、ロード処理の投機実行を行っても良いならば、コード効率を上げるとともにプログラムを簡素化できる。

#### 【 0 2 1 5 】

上記実施の形態 1 から実施の形態 6 による情報処理装置には多くの変形例があり得る。この発明による、リピート処理途中でブレイク（命令処理シーケンス切り替え）をハードウェアにより実現する技術は、上記実施の形態に開示している情報処理装置に限定的に適用されるものではなく、リピート処理を行うどのような形態の情報処理装置であっても適用可能であり、同様の効果を得ることができる。例えば、情報処理装置は V L I W プロセッサでなく、スパースケラを一含む通常のプロセッサや D S P 等のようなプロセッサであってもよい。また、この発明は上記したようなパイプライン構成や命令実行制御方法等に限定的に適用されるものではない。

#### 【 0 2 1 6 】

さらに、この発明による情報処理装置は、上述の実施の形態に示したようなリピート関連の設定を効率よく行う専用のリピート命令を用いる代わりに、プリミティブとなる命令でリピート関連のハードウェアリソースの初期設定を行うようにしてもよい。また、上述の実施の形態による情報処理装置は、実行する命令で指定されるオペレーションとは独立してハードウェア的にリピート制御を行うものであるが、通常に分岐命令やループ制御命令（ループカウンタ値のデクリメント、カウンタ値の判定を伴う分岐／ジャンプ命令等）でループ処理を実現するものにも適用可能である。

## 【 0 2 1 7 】

また、ハードウェアによるリピート制御の実現方法を、搭載するハードウェア量（コスト）と性能向上のオーバーヘッドを考慮して決定することが可能である。リピートブロック内で実行できる命令や処理可能な条件（リピートブロック内の実行命令数や繰り返し回数など）等に制約を加えてハードウェア量を削減してもよいし、性能優先でハードウェアを追加して限定を少なくなるような選択をしてもよい。リピートブロックのサイズに制限を加えるのであれば、アドレスの比較等についても全ビットで比較する必要はなく、リピートブロックの最大サイズに相当するビット数分のみ比較するようにして、ハードウェアコストを削減してもよい。

## 【 0 2 1 8 】

また、この発明による特徴の機能を実現するためにどのような実現手段を用いてもかまわない。例えば、RPT\_C等のカウント値についても、設定値からデクリメントする代わりに、ゼロからインクリメントするように更新するように制御し、そのカウント値と設定値との比較を行うような実現手段を用いてもよい。

## 【 0 2 1 9 】

また、アドレス比較、カウント値の先行更新／判定、命令処理シーケンスの切り替えを、命令フェッチ段階で行う代わりに命令実行段階で行ってもよい。この場合においてもコードサイズを削減することができる。また、この場合、命令フェッチ段階で判定するよりもハードウェアコストを削減できる。ただし、命令実行段階でのシーケンス切り替えとなるため、分岐によるオーバーヘッドが生じるため、性能の向上度は低くなる。

## 【 0 2 2 0 】

上記したように、上記したいずれの実施の形態を採用しても、リピート処理途中のブレイク（命令処理シーケンスの切り替え）をハードウェアにより実現しているので、繰り返し回数がダイナミックに変化するとともにリピートブロックの途中でリピート処理を終了する場合において、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる。従って、プログラムをROM化する場合

合のコストも削減できるうえに、プログラム開発の生産性を向上することができる。

#### 【 0 2 2 1 】

##### 【発明の効果】

以上のように、この発明によれば、情報処理装置が、実行される命令で特定されるオペレーションとは独立して、リピートブロック内で繰り返し処理の終了を検出する、ハードウェアにより実現された検出手段と、上記検出手段が繰り返し処理の終了を検出した場合、上記リピートブロック内の残りの命令の実行を抑止する命令実行抑止手段とを備えるので、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる効果がある。従って、プログラムをROM化する場合のコストも削減できるうえに、プログラム開発の生産性を向上することができる効果がある。

#### 【 0 2 2 2 】

この発明によれば、命令実行抑止手段が、検出手段が繰り返し処理の終了を検出した場合、リピートブロック内の残りの命令をノーオペレーション化する、ハードウェアにより実現された手段であるので、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる効果がある。また、単にリピートブロック内の残りの命令をノーオペレーション化するので、ハードウェアコストをさらに低減できる効果がある。

#### 【 0 2 2 3 】

この発明によれば、命令実行抑止手段が、検出手段が繰り返し処理の終了を検出した場合、命令フェッチ段階でリピートブロックの次の命令に命令処理シーケンスを切り替える命令処理シーケンス切替手段であるので、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる効果がある。

#### 【 0 2 2 4 】

この発明によれば、命令実行抑止手段が、検出手段が繰り返し処理の終了を検

出した場合、命令実行段階でリピートブロックの次の命令に命令処理シーケンスを切り替える命令処理シーケンス切替手段であるので、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる効果がある。

## 【 0 2 2 5 】

この発明によれば、命令処理シーケンス切替手段が、リピートブロックの繰り返し処理において最後に実行される最終命令の実行時に、リピートブロックの次の命令へのジャンプ処理を行う手段であるので、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる効果がある。

## 【 0 2 2 6 】

この発明によれば、命令処理シーケンス切替手段が、リピートブロックの繰り返し処理において最後に実行される最終命令の実行後に、リピートブロックの次の命令へのジャンプ処理を行う手段であるので、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる効果がある。

## 【 0 2 2 7 】

この発明によれば、検出手段が、リピートブロックの繰り返し処理時に実行される命令のアドレスに基づき繰り返し処理が終了したか否かを判定する手段であるので、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかることができるとともに、プログラムサイズを大幅に削減できる効果がある。

## 【 0 2 2 8 】

この発明によれば、検出手段が、リピートブロックを繰り返し処理する回数をカウントするカウント手段と、上記リピートブロック内の実行される命令のアドレスと最後に実行される最終命令のアドレスとを比較する比較手段とを有しており、上記カウント手段のカウント数が所定の数に到達し、上記比較手段が実行される命令のアドレスが最終命令のアドレスと一致したことを通知した際に繰り返し処理の終了と判定するので、条件判定、分岐のための前処理／後処理によるオ

オーバーヘッドをなくし性能の向上をはかるとともに、プログラムサイズを大幅に削減できる効果がある。

【0229】

この発明によれば、検出手段が、リピートブロックの繰り返し処理時に実行される命令数に基づき繰り返し処理が終了したか否かを判定する手段であるので、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかるとともに、プログラムサイズを大幅に削減できる効果がある。

【0230】

この発明によれば、検出手段が、リピートブロック内の実行された命令数をカウントするカウント手段を有しており、上記カウント手段のカウント数が所定の数に到達した際に繰り返し処理の終了と判定するので、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかるとともに、プログラムサイズを大幅に削減できる効果がある。

【0231】

この発明によれば、検出手段が、リピートブロックを繰り返し処理する回数をカウントする第1カウント手段と、上記リピートブロック内の実行された命令数をカウントする第2カウント手段とを有しており、上記第1カウント手段のカウント数が第1の所定の数に到達し、上記リピートブロックの最終の繰り返し処理において上記第2カウント手段のカウント数が第2の所定の数に到達した際に繰り返し処理の終了と判定するので、条件判定、分岐のための前処理／後処理によるオーバーヘッドをなくし性能の向上をはかるとともに、プログラムサイズを大幅に削減できる効果がある。

【図面の簡単な説明】

【図1】 この発明の実施の形態1による情報処理装置のレジスタセットを示す図である。

【図2】 制御レジスタCROに保持されるPSWの構成を示す図である。

【図3】 この発明による情報処理装置の32ビット命令のフォーマットを示す図である。

【図 4】 2つのオペランドを持つショート命令のビット割り付けの例を示す図である。

【図 5】 ショートフォーマットの分岐命令のビット割り付けの例を示す図である。

【図 6】 16ビットの変位や即値を持つ3オペランド命令やロード／ストア命令のビット割り付けの例を示す図である。

【図 7】 右コンテナ側にオペレーションコードを持つロングフォーマットの命令のビット割り付けの例を示す図である。

【図 8】 この発明の実施の形態1による情報処理装置の構成を示すブロック図である。

【図 9】 図8に示すこの発明の実施の形態1による情報処理装置の第1演算部の構成を詳細に示すブロック図である。

【図 10】 図8に示すこの発明の実施の形態1による情報処理装置のPC部の構成を詳細に示すブロック図である。

【図 11】 図8に示すこの発明の実施の形態1による情報処理装置の第2演算部の構成を詳細に示すブロック図である。

【図 12】 この発明の実施の形態1による情報処理装置のパイプライン処理を示す図である。

【図 13】 ロードオペランド干渉の例を示す図である。

【図 14】 演算ハードウェア干渉の例を示す図である。

【図 15】 リピート命令“REP Rsrc1, Rsrc2, displ 6”のビット割り付けを示す図である。

【図 16】 積和演算を行うアセンブラで記述されており、REP命令を含むプログラム例を示す図である。

【図 17】 この発明の実施の形態1による、図16に示したプログラムのREP命令実行からリピート処理完了までのパイプライン処理を示すタイミングチャート図である。

【図 18】 この発明の実施の形態1による、図16に示したプログラムのREP命令実行からリピート処理完了までのパイプライン処理を示すタイミング

チャート図である。

【図 1 9】 リピートブロック内の 1 命令しか実行しない場合の処理例を示すタイミングチャート図である。

【図 2 0】 この発明の実施の形態 2 による情報処理装置が用いる P S W の構成を示す図である。

【図 2 1】 この発明の実施の形態 2 による情報処理装置の P C 部の構成を示すブロック図である。

【図 2 2】 この発明の実施の形態 2 による、図 1 6 に示したプログラム実行時のパイプライン処理を示すタイミングチャート図である。

【図 2 3】 この発明の実施の形態 3 による、図 1 6 に示したプログラム実行時のパイプライン処理を示すタイミングチャート図である。

【図 2 4】 この発明の実施の形態 4 による、図 1 6 に示したプログラム実行時のパイプライン処理を示すタイミングチャート図である。

【図 2 5】 この発明の実施の形態 5 による情報処理装置のリピート関連の制御レジスタを示す図である。

【図 2 6】 この発明の実施の形態 5 による情報処理装置の P C 部の構成を示すブロック図である。

【図 2 7】 この発明の実施の形態 5 による、図 1 6 に示したプログラム実行時のパイプライン処理を示すタイミングチャート図である。

【図 2 8】 この発明の実施の形態 5 による、図 1 6 に示したプログラム実行時のパイプライン処理を示すタイミングチャート図である。

【図 2 9】 この発明の実施の形態 6 による情報処理装置のリピート関連の制御レジスタを示す図である。

【図 3 0】 この発明の実施の形態 6 による情報処理装置の P C 部の構成を示すブロック図である。

【図 3 1】 この発明の実施の形態 6 による情報処理装置のディレイドリピート（遅延リピート）命令 “D R E P R s r c , d i s p l 6” の命令フォーマットを示す図である。

【図 3 2】 この発明の実施の形態 6 による情報処理装置のディレイドリピ



ート命令 “D R E P I   i m m 8 ,   d i s p 1 6” の命令フォーマットを示す図である。

【図 3 3】   積和演算を行うアセンブラで記述されており、D R E P 命令を含むプログラム例を示す図である。

【図 3 4】   この発明の実施の形態 6 による情報処理装置のリピート処理中のパイプライン処理を示すタイミングチャート図である。

【図 3 5】   この発明の実施の形態 6 による情報処理装置のリピート処理中のパイプライン処理を示すタイミングチャート図である。

【図 3 6】   従来の情報処理装置の繰り返し処理の例を示すフローチャートである。

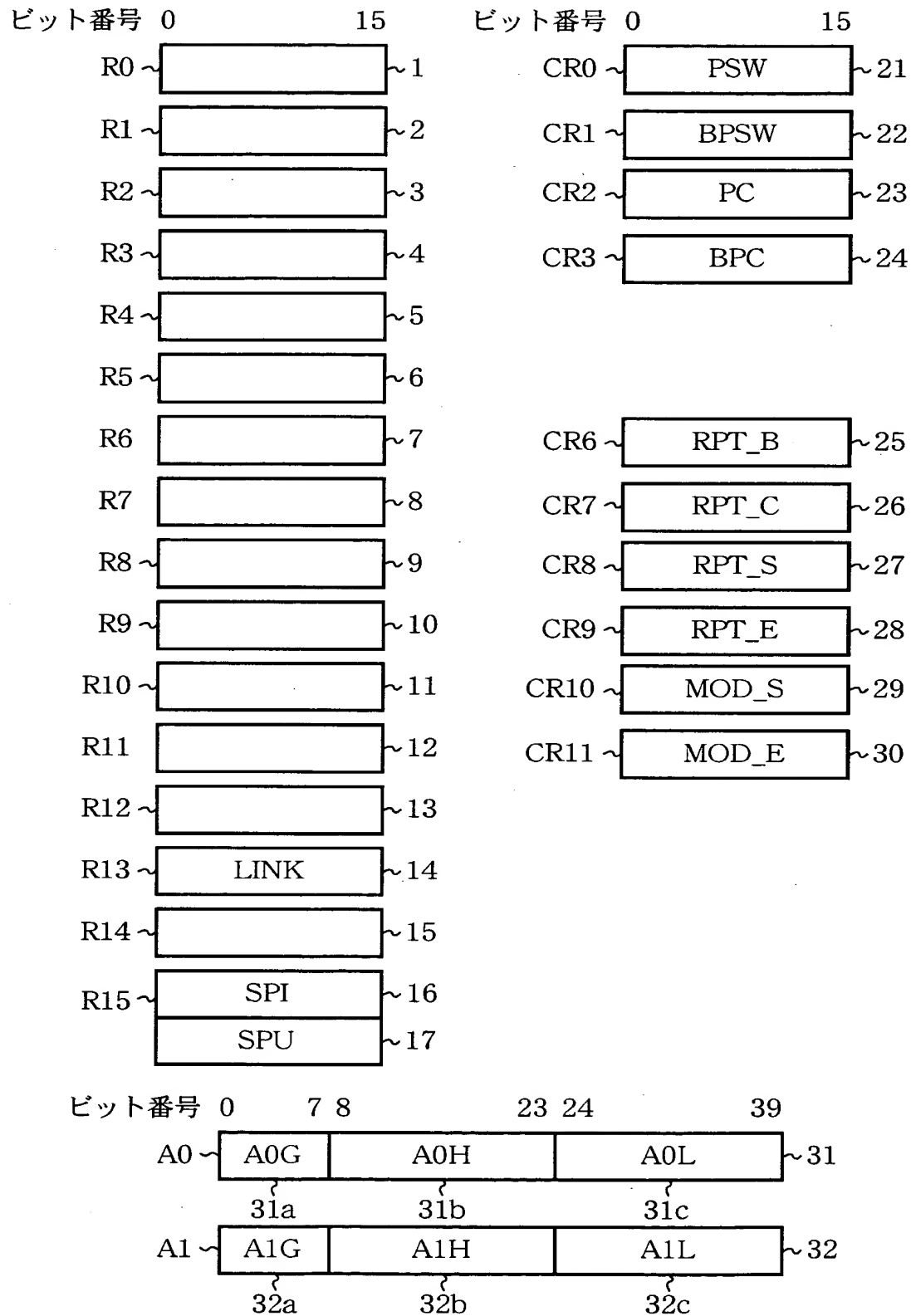
#### 【符号の説明】

1 ~ 1 7   汎用レジスタ、2 1 ~ 3 0 ,   7 0 1 ,   8 0 1   制御レジスタ、3 1 ,   3 2   アキュムレータ、4 1   S M ビット、4 2   I E ビット、4 3   R P ビット、4 4   M D ビット、4 5   F X ビット、4 6   S T ビット、4 7 ,   4 8   実行制御フラグ、4 9   キャリーフラグ、5 1 ,   5 0 1 ,   8 3 1 ,   8 3 6   フォーマット指定ビット ( F M ビット ) 、 5 2   左コンテナ、5 3   右コンテナ、6 1 ,   6 4 ,   7 1 ,   8 1 ,   9 3 ,   9 6   オペレーションコードフィールド、6 2 ,   6 3 ,   8 2 ,   8 3 ,   9 1 ,   9 4 ,   9 5   フィールド、7 2   分岐変位フィールド、8 4   拡張データフィールド、9 2   予約フィールド、1 0 0   情報処理装置、1 0 1   M P U コア部、1 0 2   命令フェッチ部、1 0 3   内蔵命令メモリ、1 0 4   オペランドアクセス部、1 0 5   内蔵データメモリ、1 0 6   外部インタフェース部、1 1 1   命令キュー、1 1 2   制御部 ( 検出手段、命令実行抑止手段、命令処理シーケンス切替手段 ) 、 1 1 3   第 1 デコーダ、1 1 4   第 2 デコーダ、1 1 5   レジスタファイル、1 1 6   第 1 演算部、1 1 7   第 2 演算部、1 1 8   P C 部 ( 検出手段、命令実行抑止手段、命令処理シーケンス切替手段 ) 、 1 1 9   命令デコード部、1 5 1   A A ラッチ、1 5 2   A B ラッチ、1 5 3 ,   2 0 1   A L U 、 1 5 4   A O ラッチ、1 5 5   セレクタ、1 5 6   M O D \_ S レジスタ、1 5 7   M O D \_ E レジスタ、1 5 8 ,   1 8 7   比較器、1 7 2   比較器 ( 比較手段 ) 、 1 5 9 ,   1 6 1 ,   1 6 3 ,   1 6 5 ,   1 6 9 ,   1 7 4 ,   1

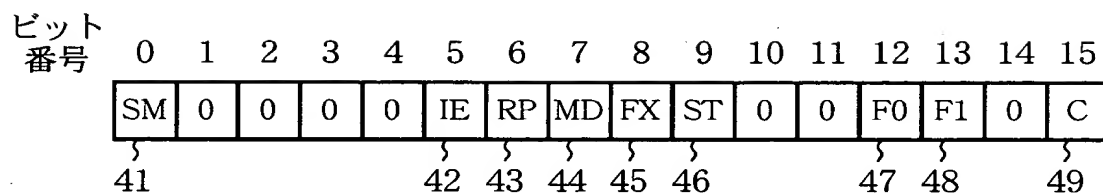
79, 182, 185, 192, 195, 712, 718, 812, 815, 818 ラッチ、160 SDレジスタ、162, 166 整置回路、164 LDレジスタ、167 TPSWラッチ、168 BPSWレジスタ、171 RPT\_Bレジスタ、173, 183, 193 インクリメンタ、175 TRPT\_Cレジスタ（カウント手段、第1カウント手段）、176 ラッチ（カウント手段、第1カウント手段）、177 デクリメンタ（カウント手段、第1カウント手段）、178, 717 ゼロ検出回路、181 IAレジスタ、184 RPT\_Sレジスタ、186 RPT\_Eレジスタ、188, 811 RPT\_Cレジスタ、191, 611 NPC、194 EPC、196 BPC、202 Aラッチ、203 Bラッチ、204, 205 シフタ、206 配線、207 マルチプレクサ、208 アキュムレータ、209 サチュレーション回路、210 プライオリティエンコーダ、211 乗算器、212 Xラッチ、213 Yラッチ、214 Pラッチ、215 バレルシフタ、216 SCラッチ、217 SDラッチ、218 即値ラッチ、221 PSW部、222 PSWラッチ、301 S1バス、302 S2バス、303 S3バス、304 S4バス、305 S5バス、311 D1バス、312 D2バス、313 D3バス、314 Wバス、321 OAバス、322 ODバス、323 JAバス、401 IFステージ、402 Dステージ、403 Eステージ、404 Mステージ、405 Wステージ、406 E2ステージ、502, 832, 837 オペレーションコード、503, 504, 834 レジスタ番号、505 リピートブロック最終命令のアドレス、601 REビット、711 RPT\_BCレジスタ、713, 813, 816 デクリメンタ、714 TRPT\_BCレジスタ（カウント手段、第2カウント手段）、715 ラッチ（カウント手段、第2カウント手段）、716 デクリメンタ（カウント手段、第2カウント手段）、814 TRPT\_Cレジスタ、817 1検出回路、833 予約ビット、838 即値データ、835, 839 リピートブロック最終命令のリピート命令からの変位値。

【書類名】 図面

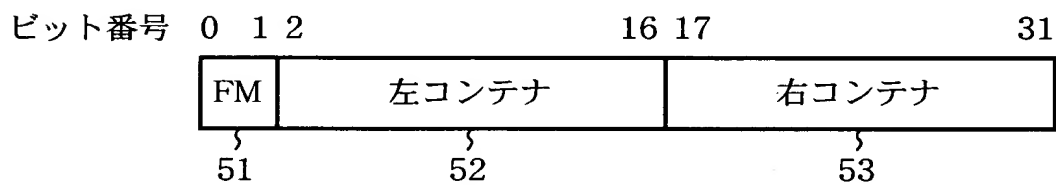
【図 1】



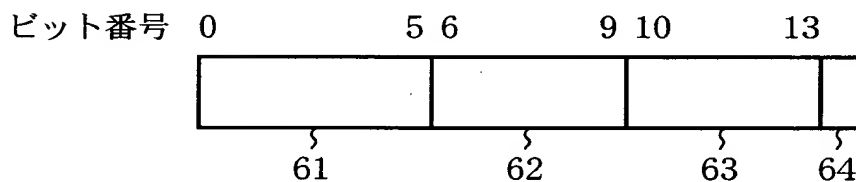
【図 2】



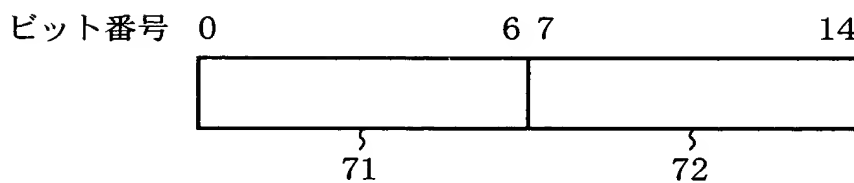
【図 3】



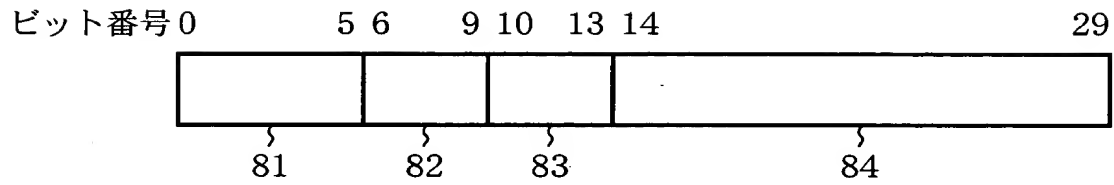
【図 4】



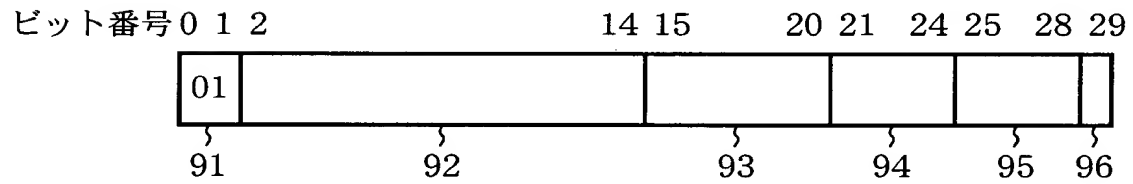
【図 5】



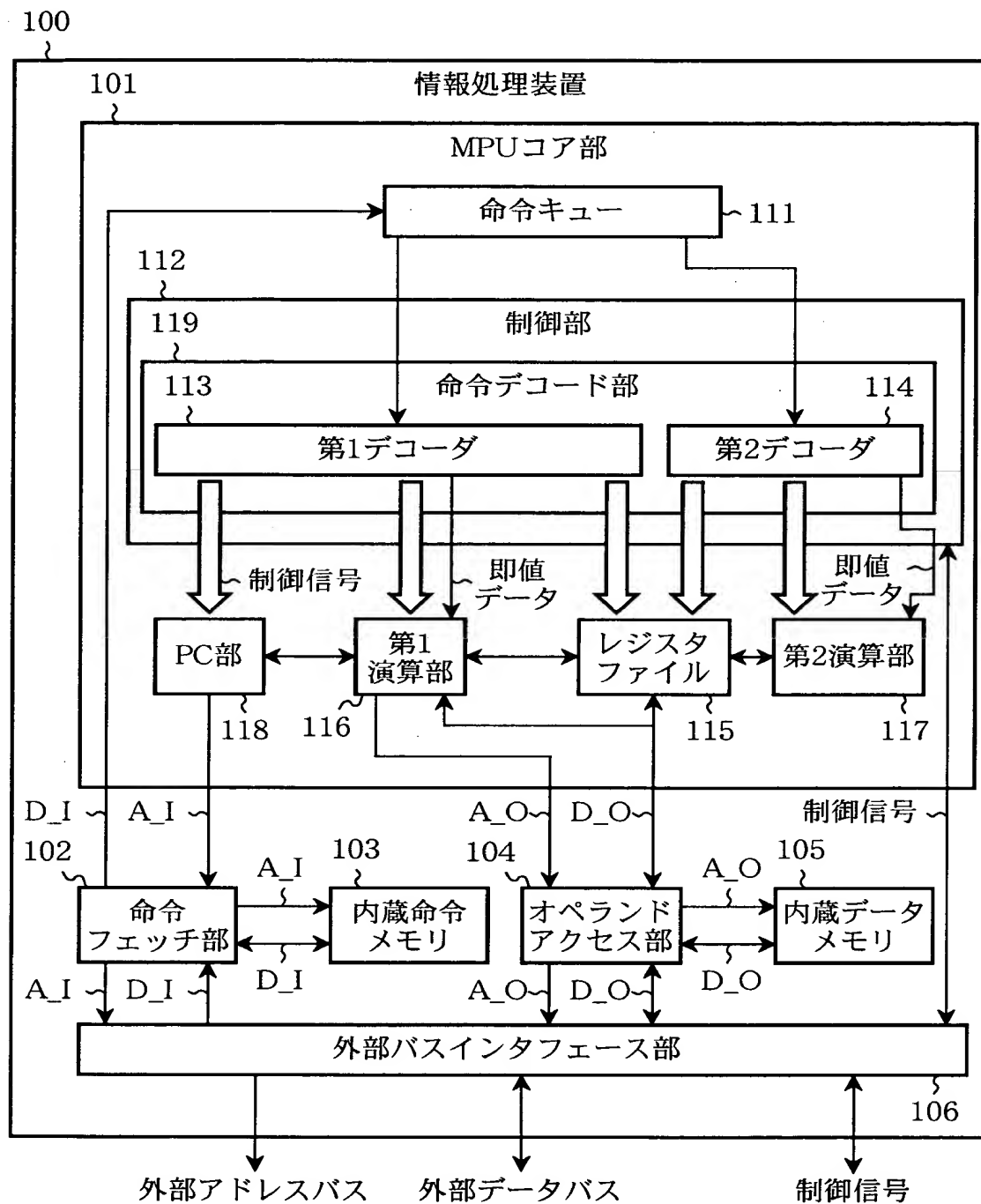
【図 6】



【図 7】

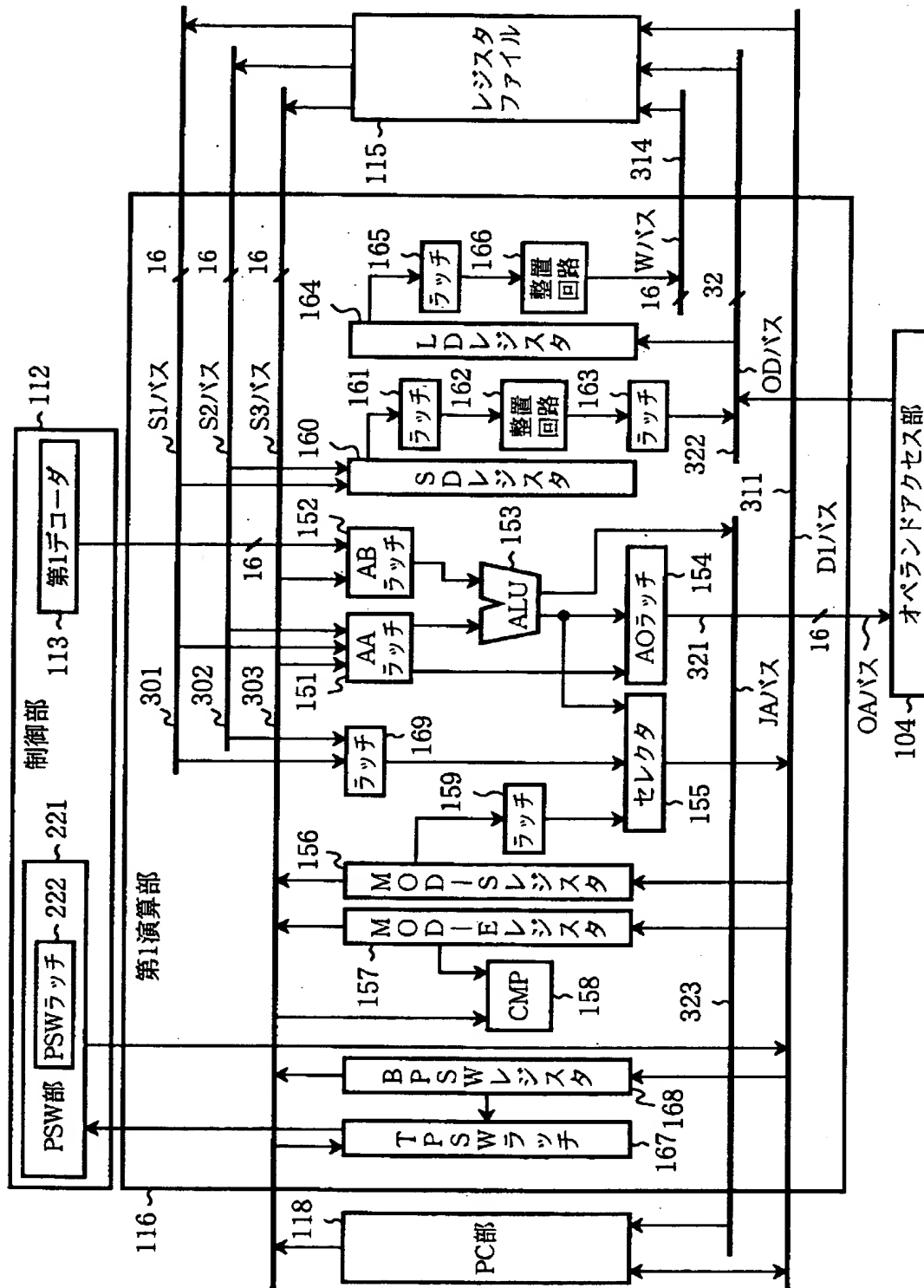


【図 8】

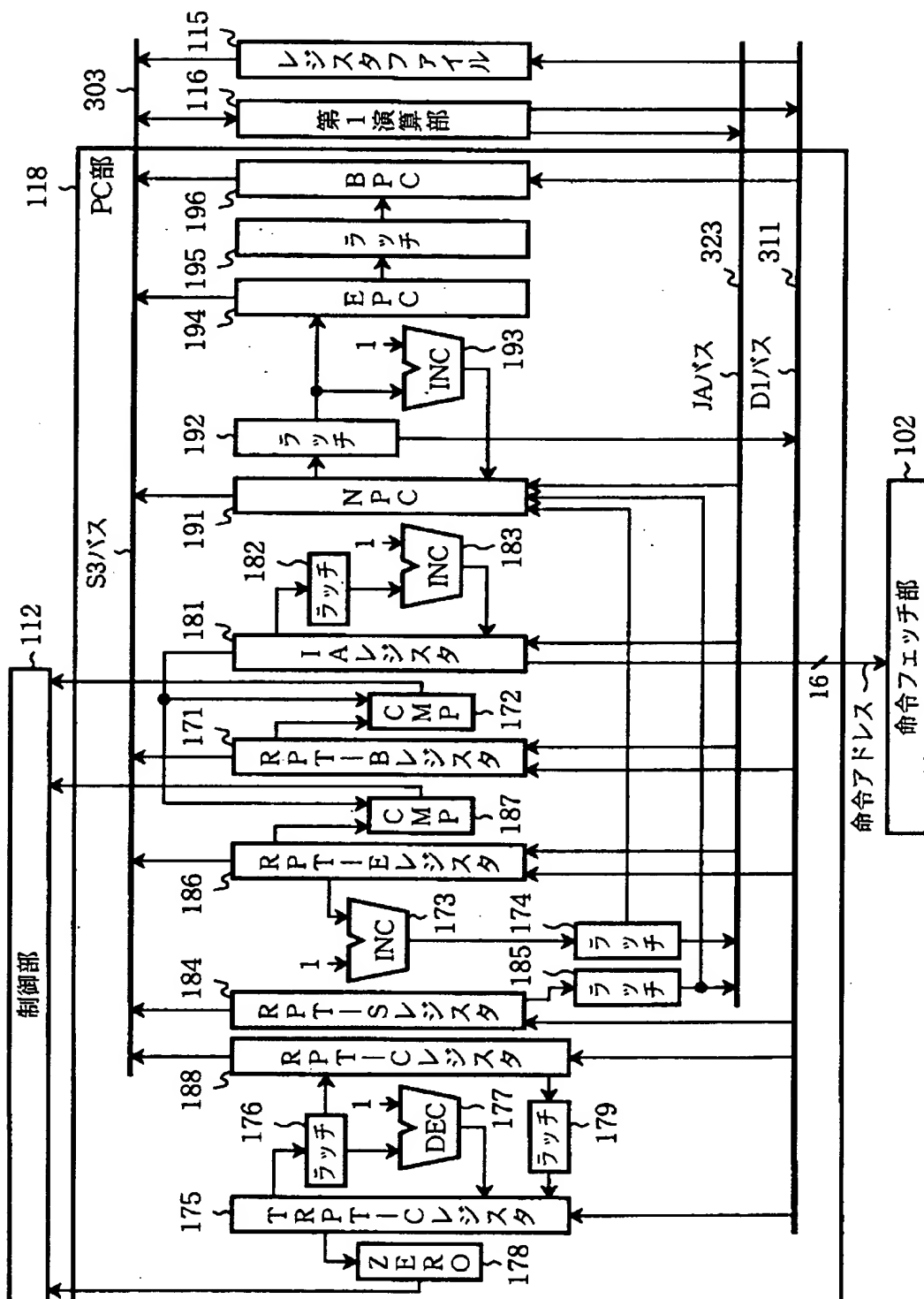


A\_I: 命令アドレス  
 D\_I: 命令データ(命令コード)  
 A\_O: オペランドアドレス  
 D\_O: オペランドデータ

【図9】

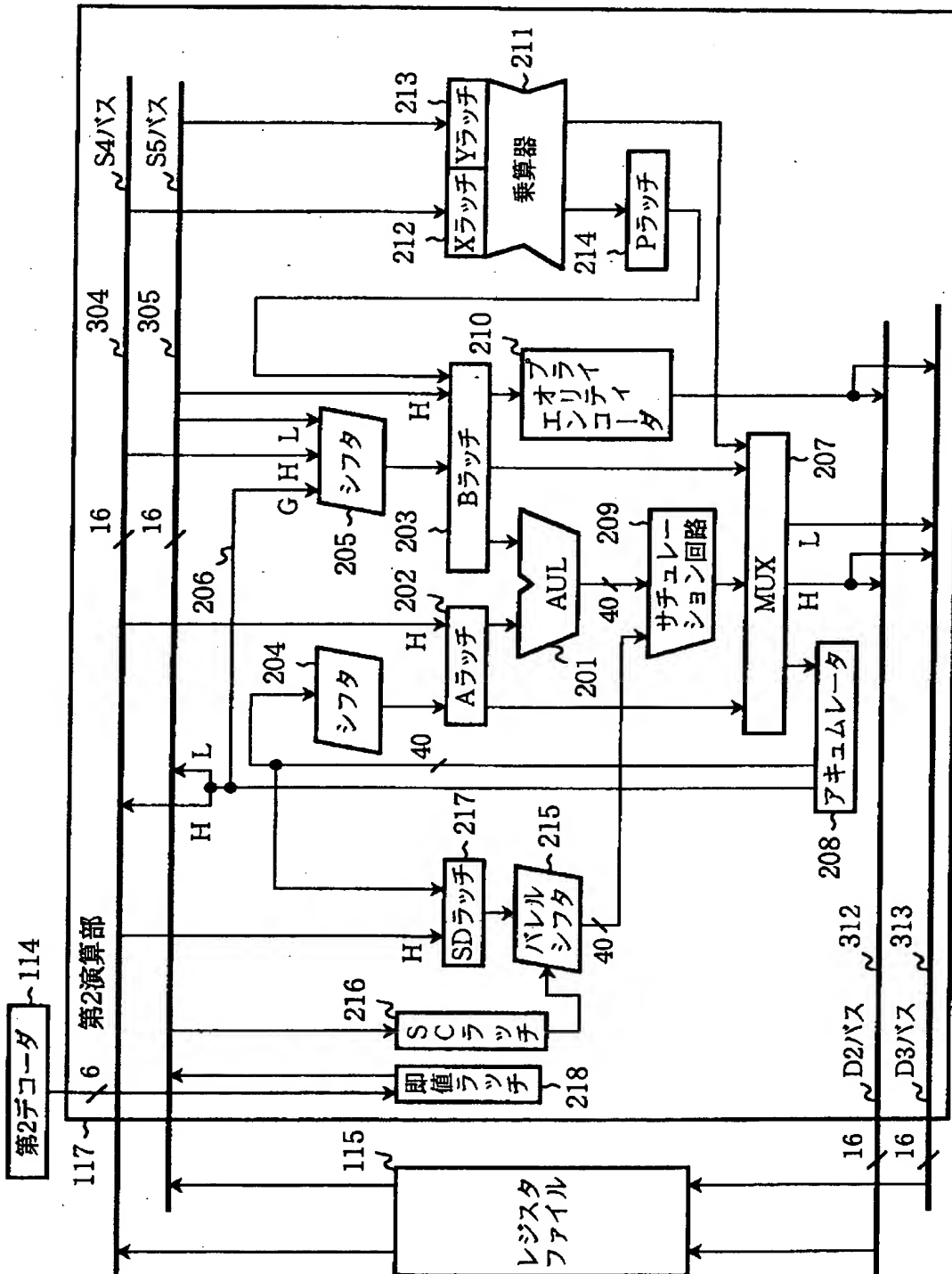


【図10】

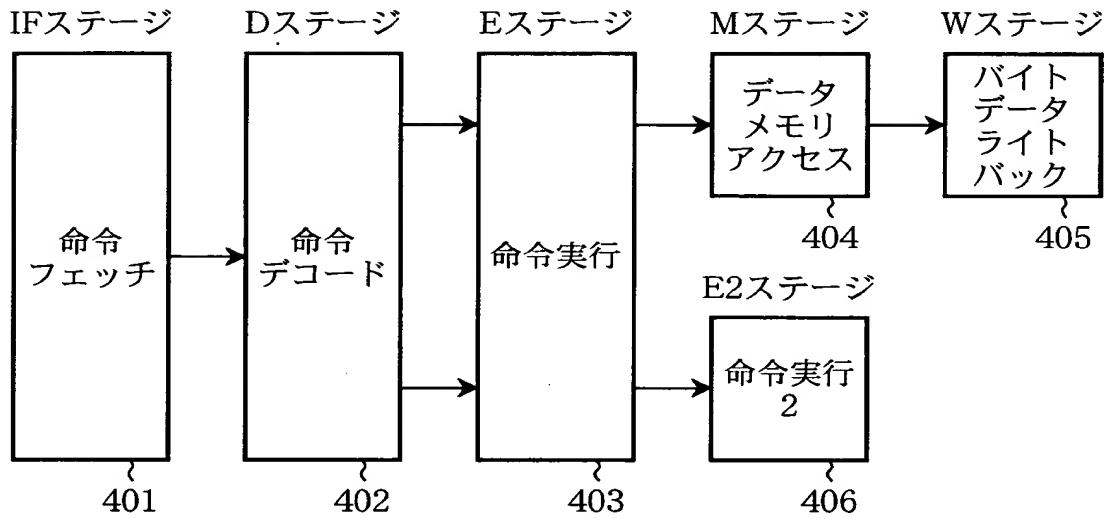




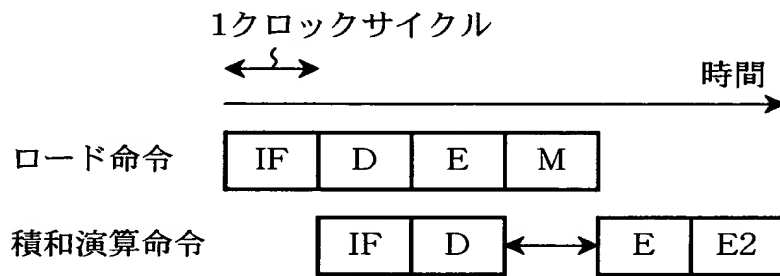
【図 11】



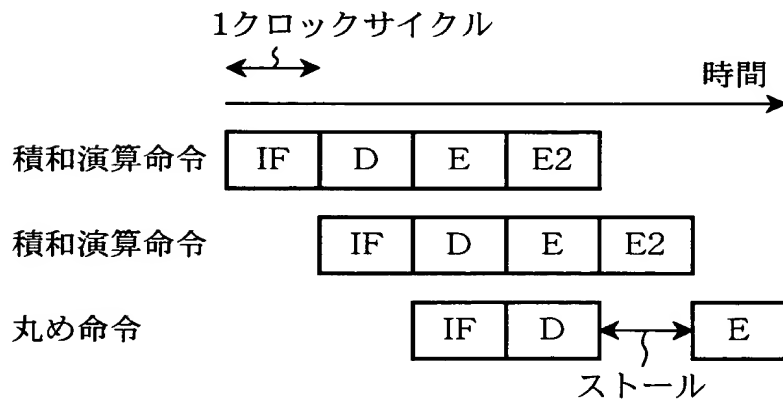
【図 1 2】



【図 1 3】



【図 1 4】



【図 15】

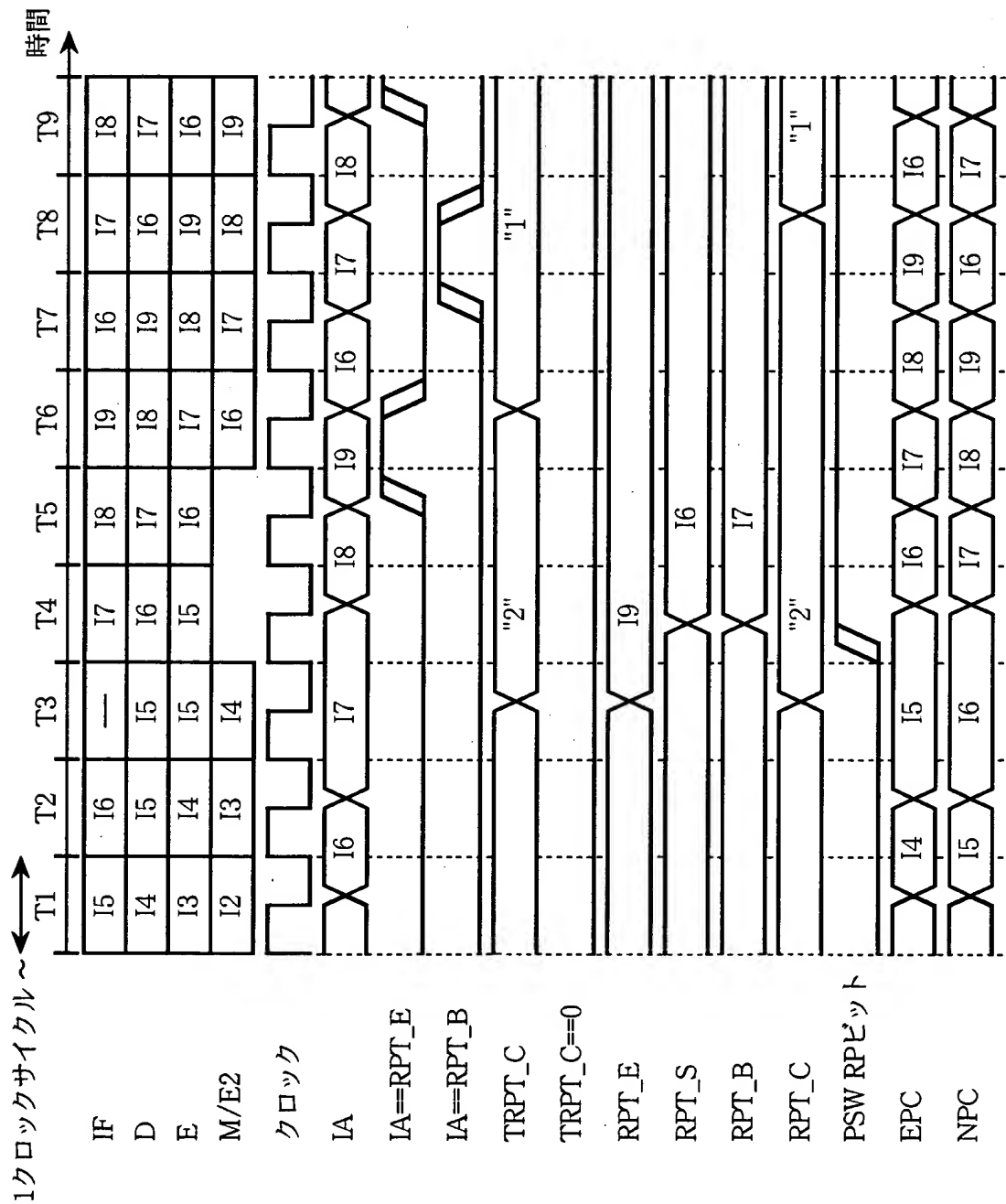
ビット番号 REP Rsrc1, Rsrc2, disp16

0 1 2			7 8		11 12		15 16		31							
11			opcode					Rsrc1		Rsrc2		disp16				
501			502					503		504		505				

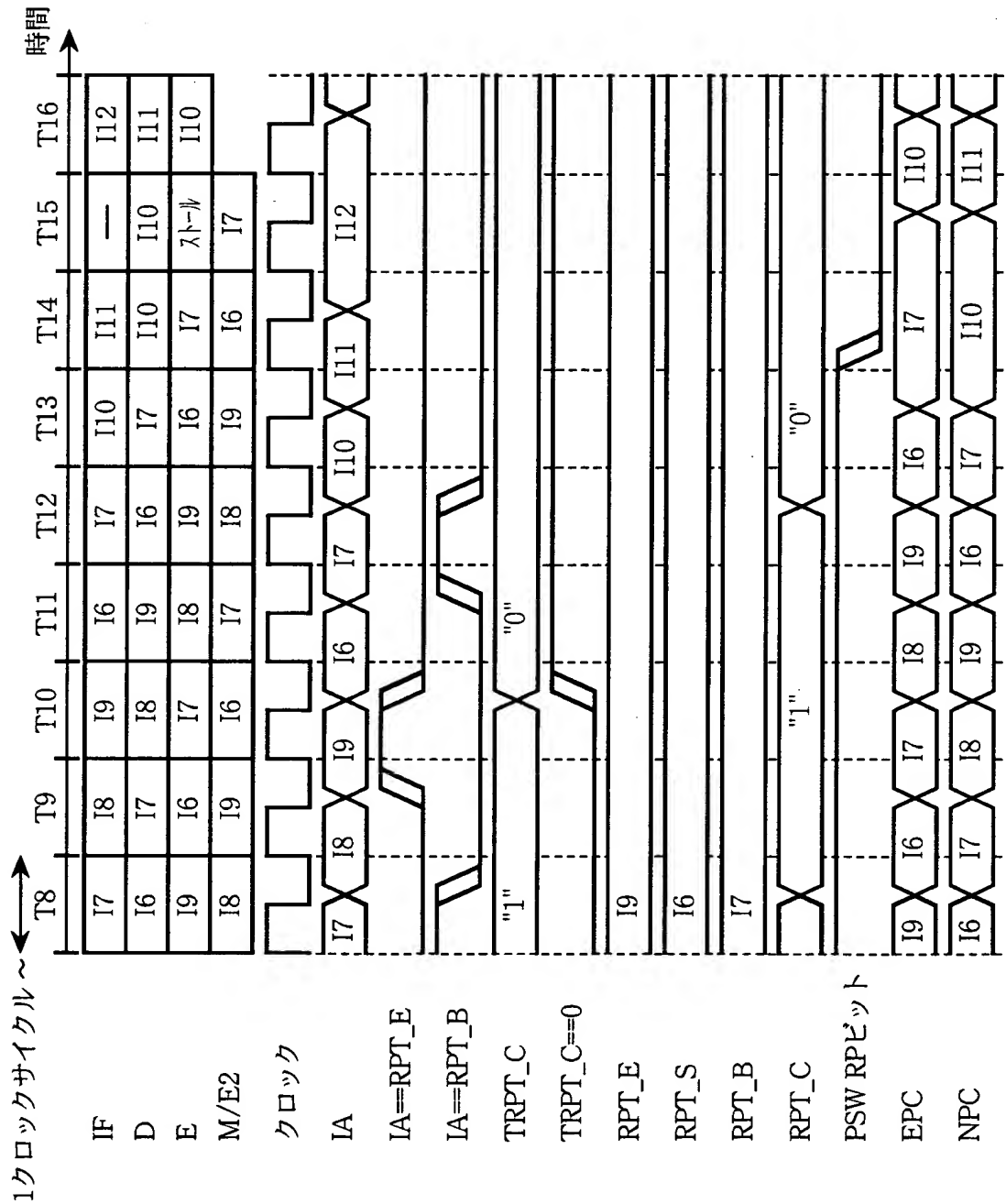
【図 16】

AND3	R11,R10,#h'0003				; I1
LD2W	R0,@R8+		SRLI	R10,#2	; I2
LD2W	R4,@R9+		NOP		; I3
LD2W	R2,@R8+		CLRAC	A0	; I4
REP	R11,R10,rep_end				; I5
LD2W	R6,@R9+		MAC	A0,R0,R4	; I6
LD2W	R0,@R8+		MAC	A0,R1,R5	; I7
LD2W	R4,@R9+		MAC	A0,R2,R6	; I8
rep_end :					
LD2W	R2,@R8+		MAC	A0,R3,R7	; I9
RACHI	R0,A0,#0		NOP		; I10
:					; I11
:					::

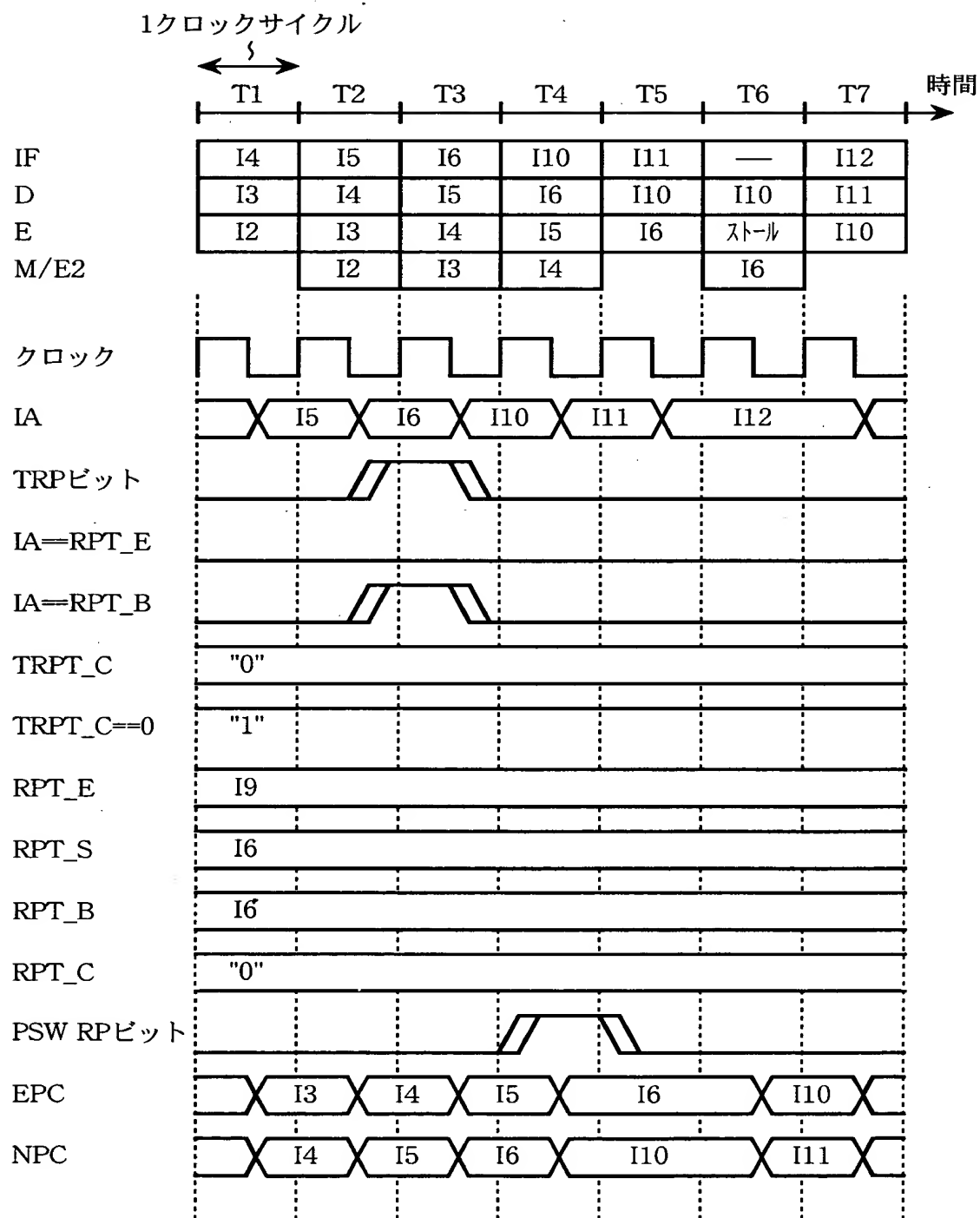
【図 17】



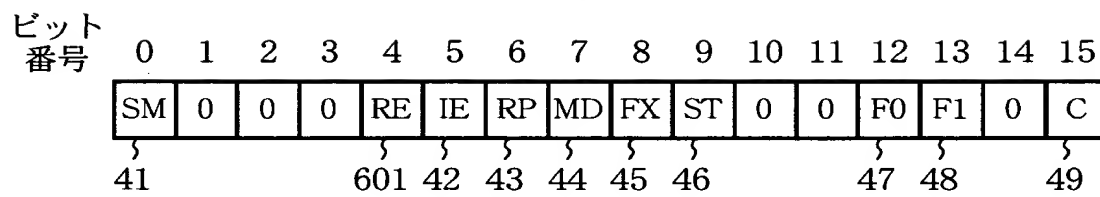
【図 18】



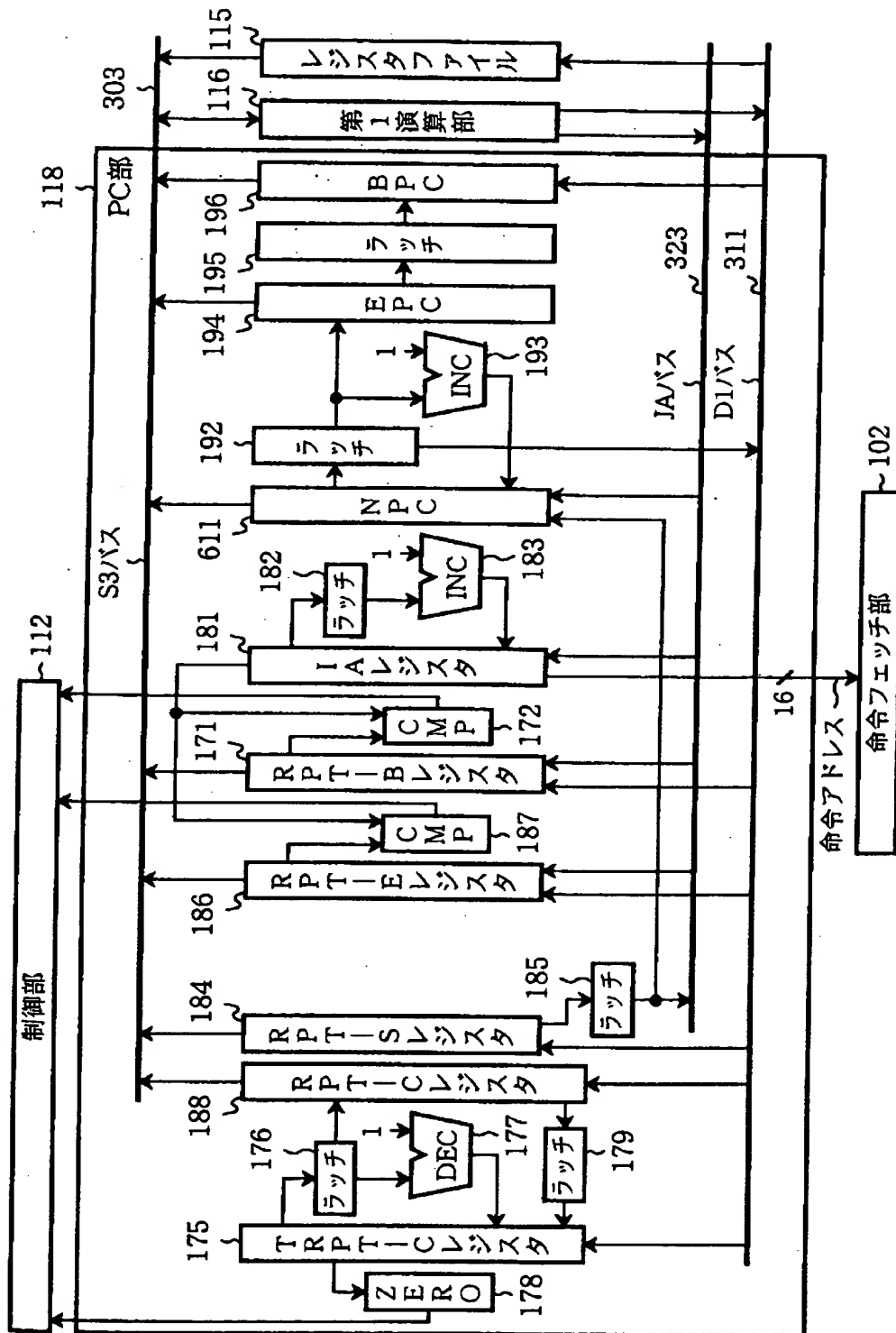
【図19】



【図 2 0】

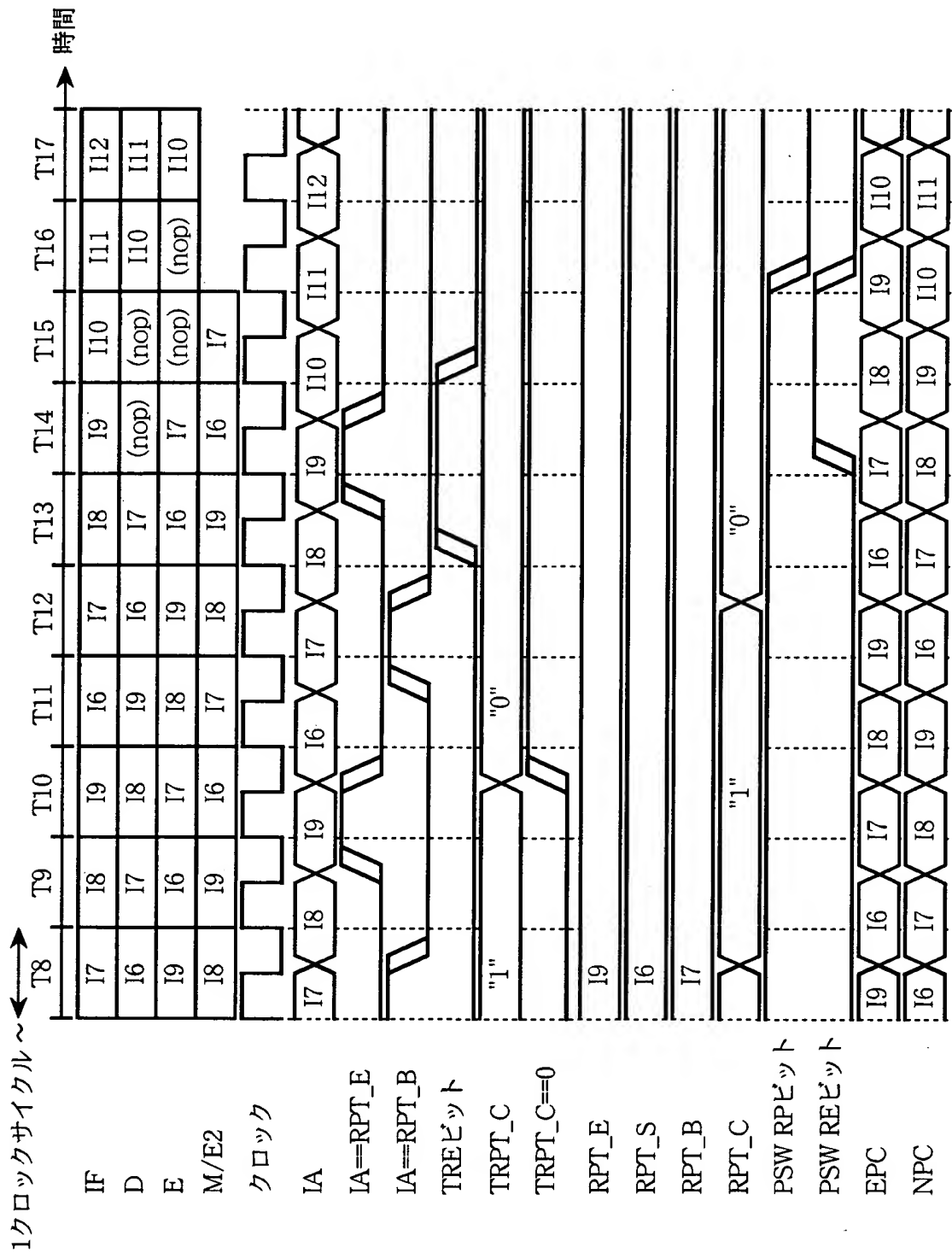


【図21】

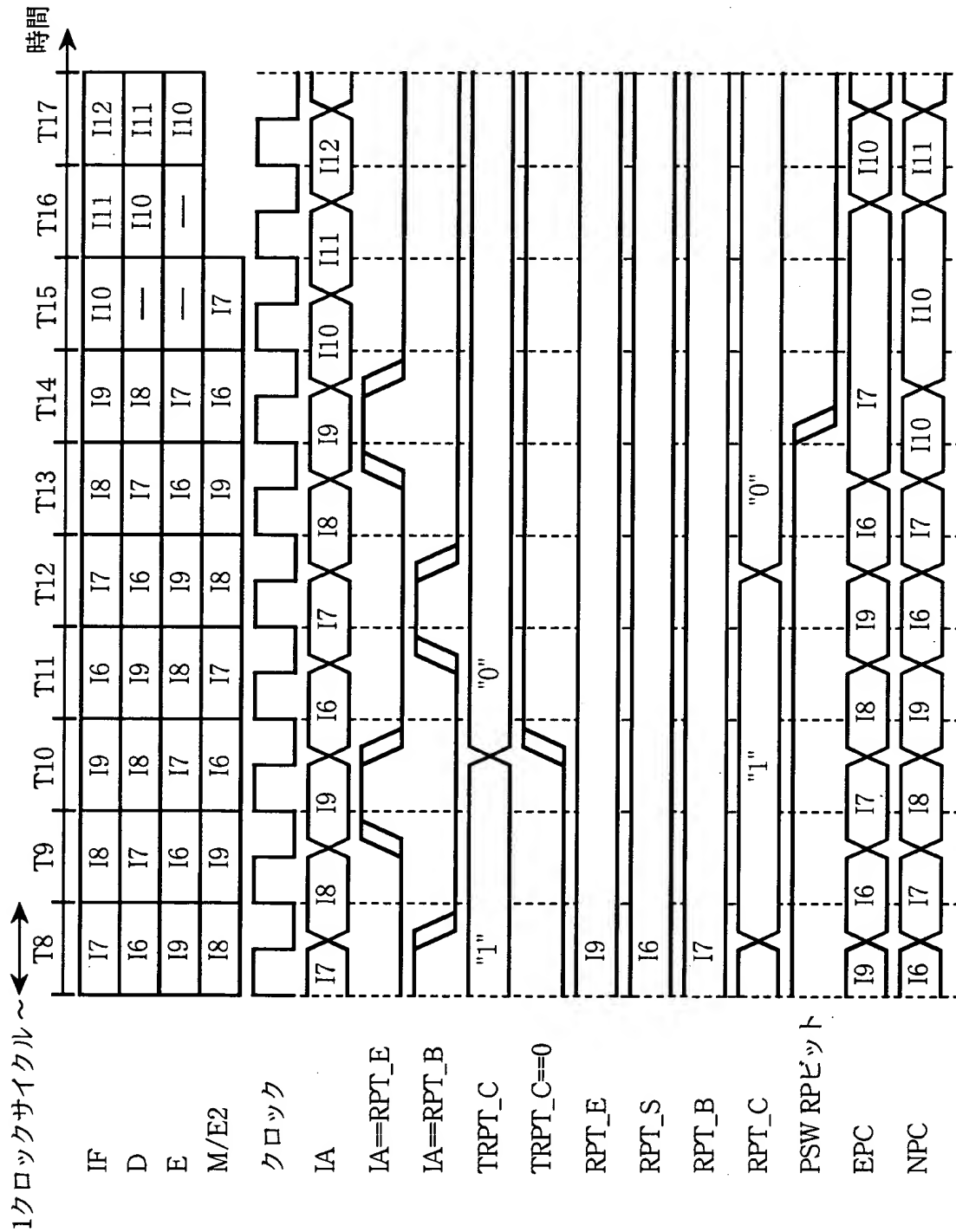




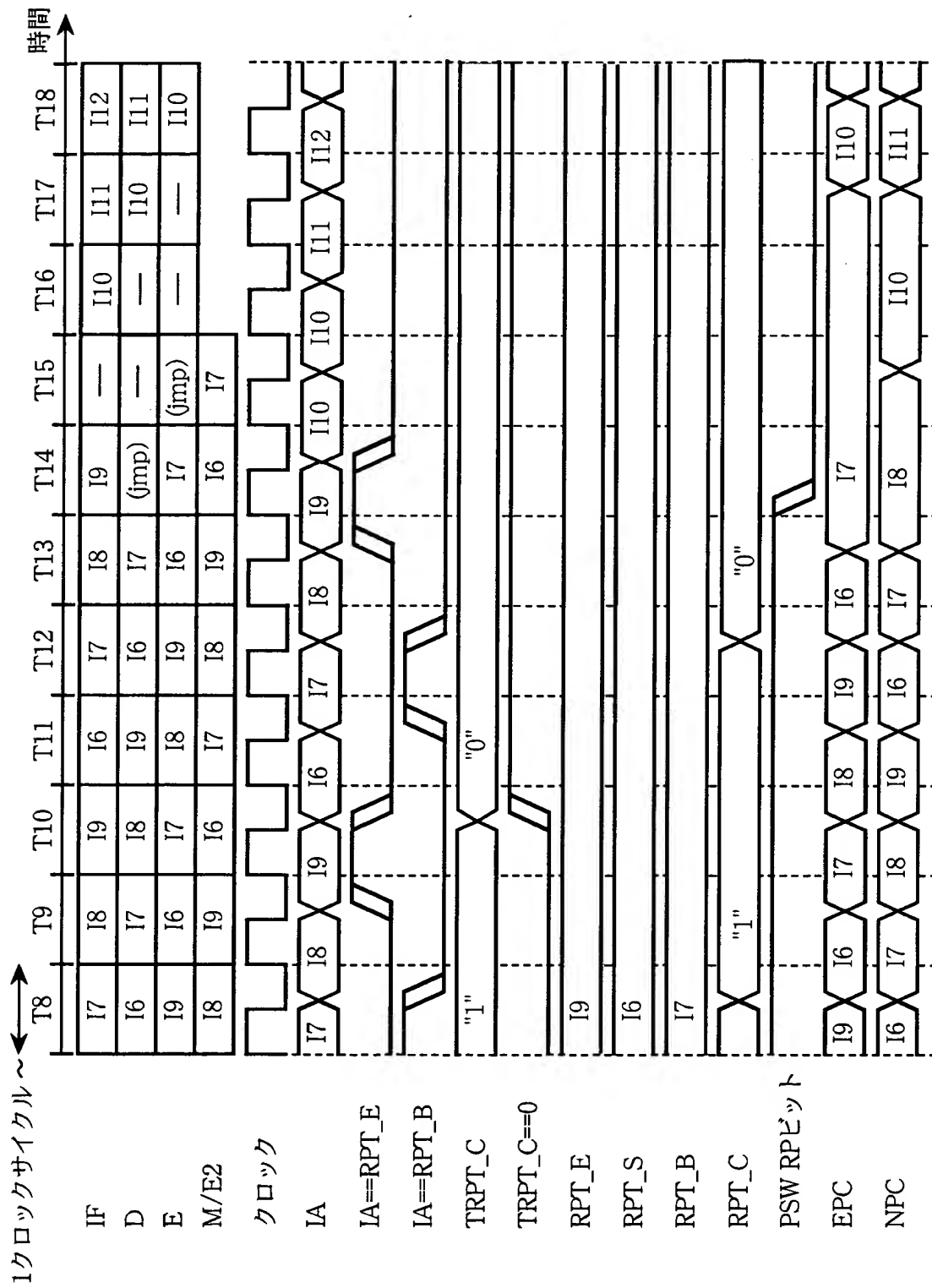
【図 22】



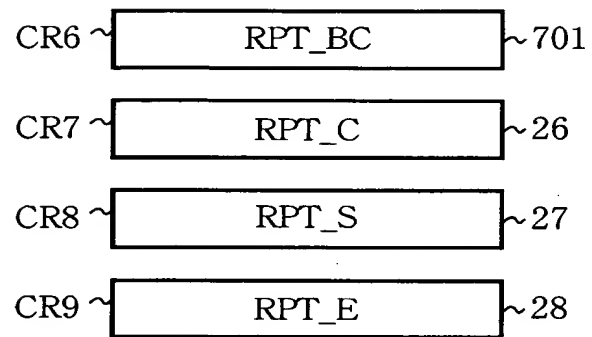
【図 23】



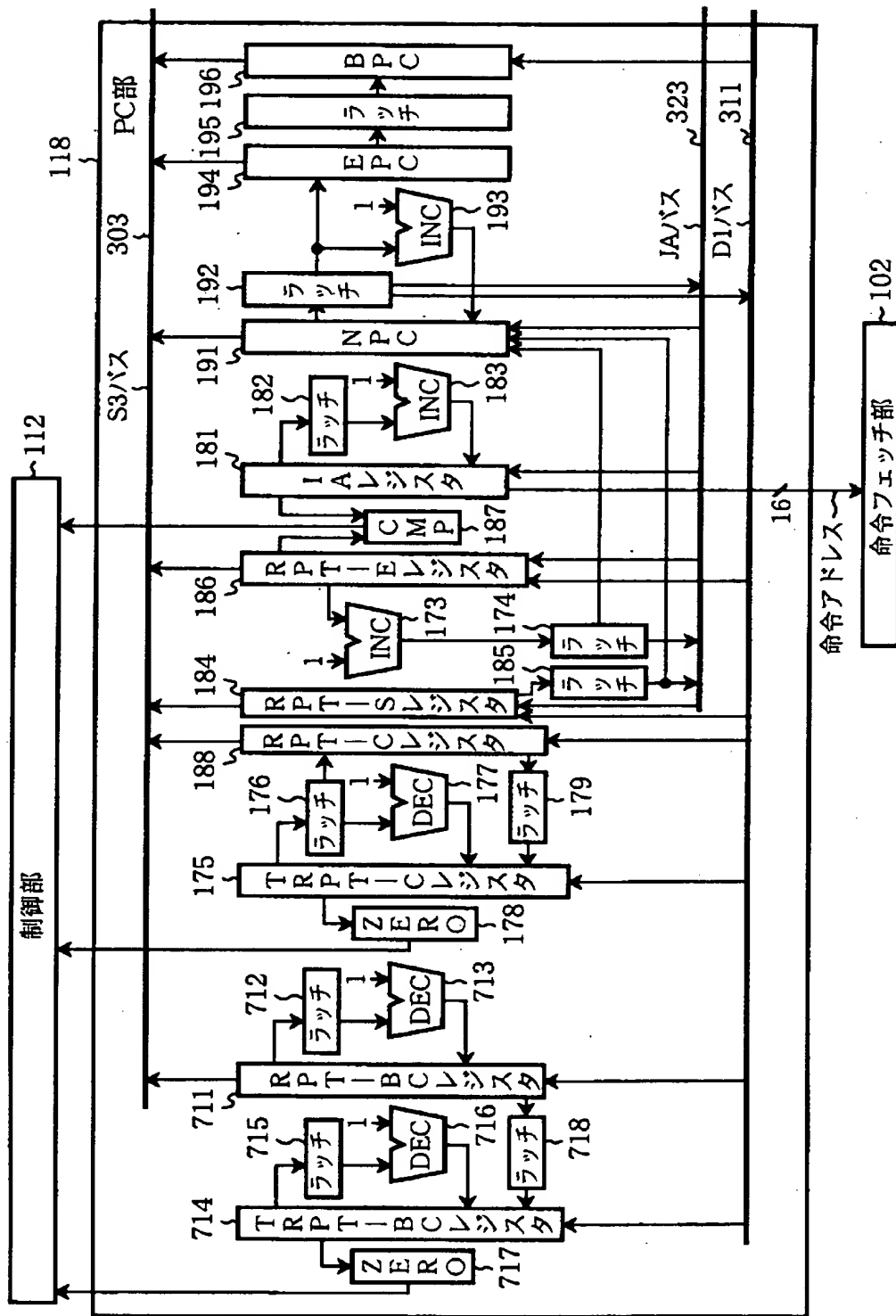
【図 24】



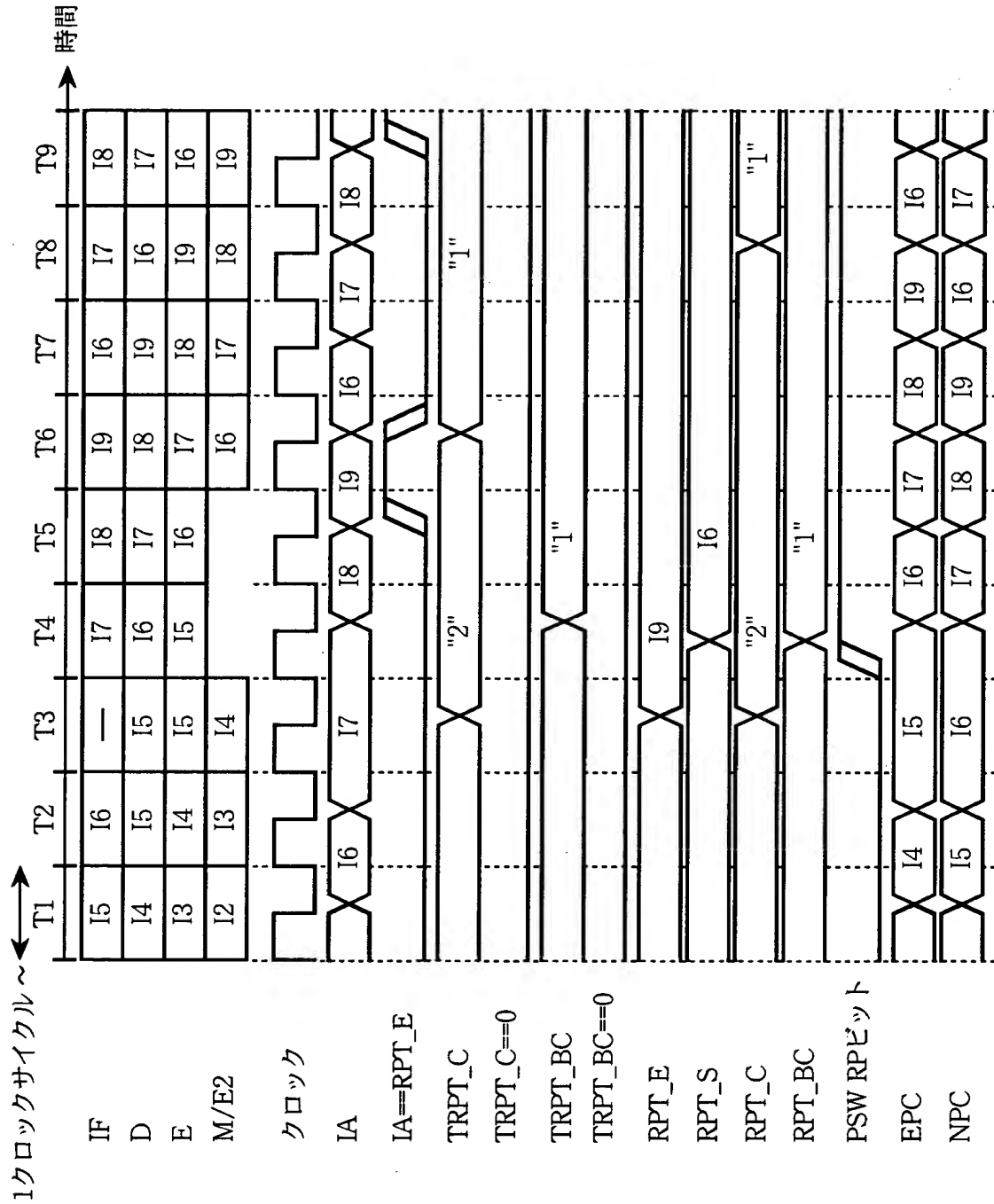
【図 2 5】



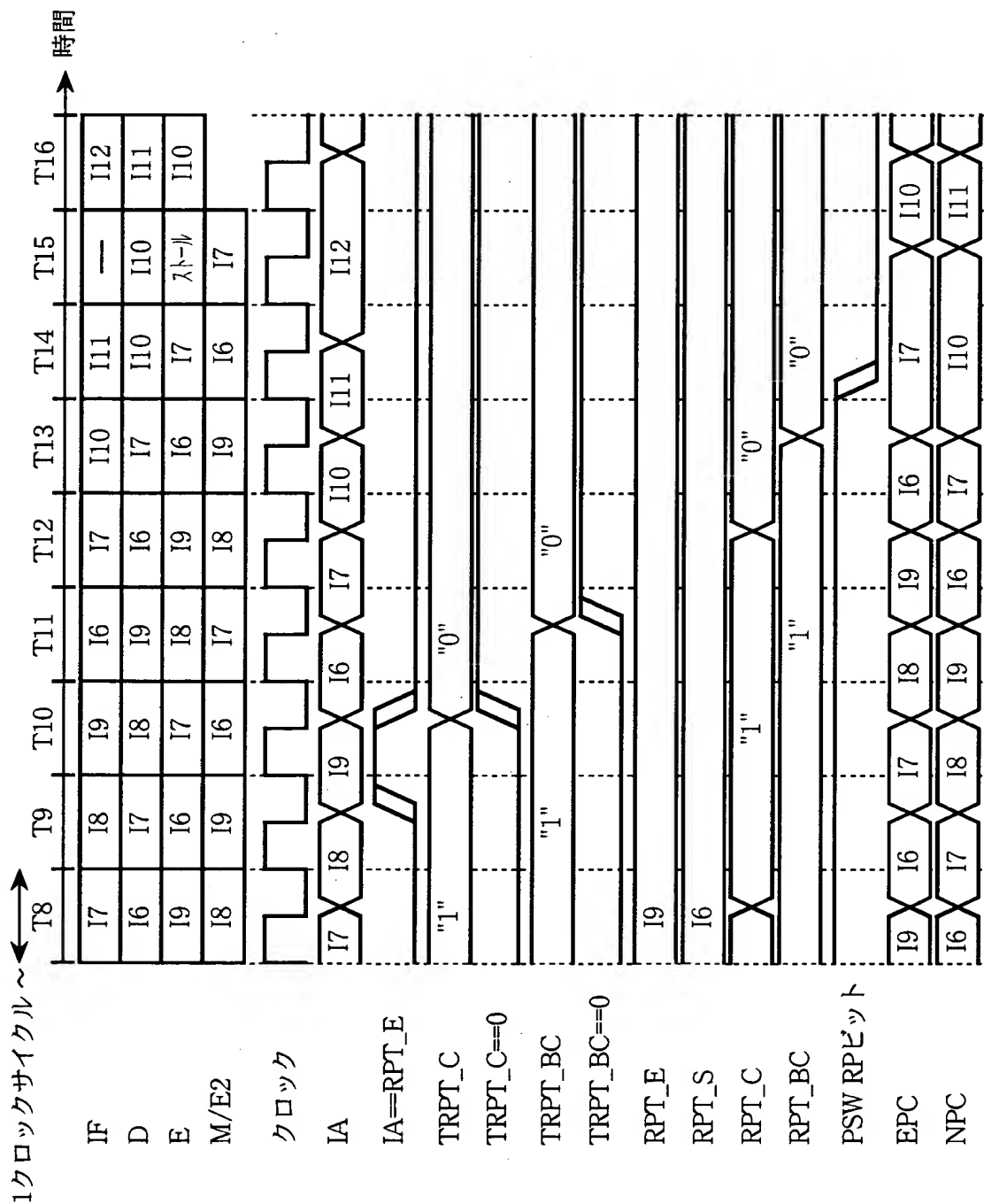
【図 26】



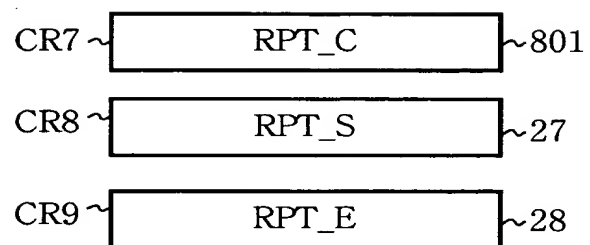
【図 27】



【图 28】

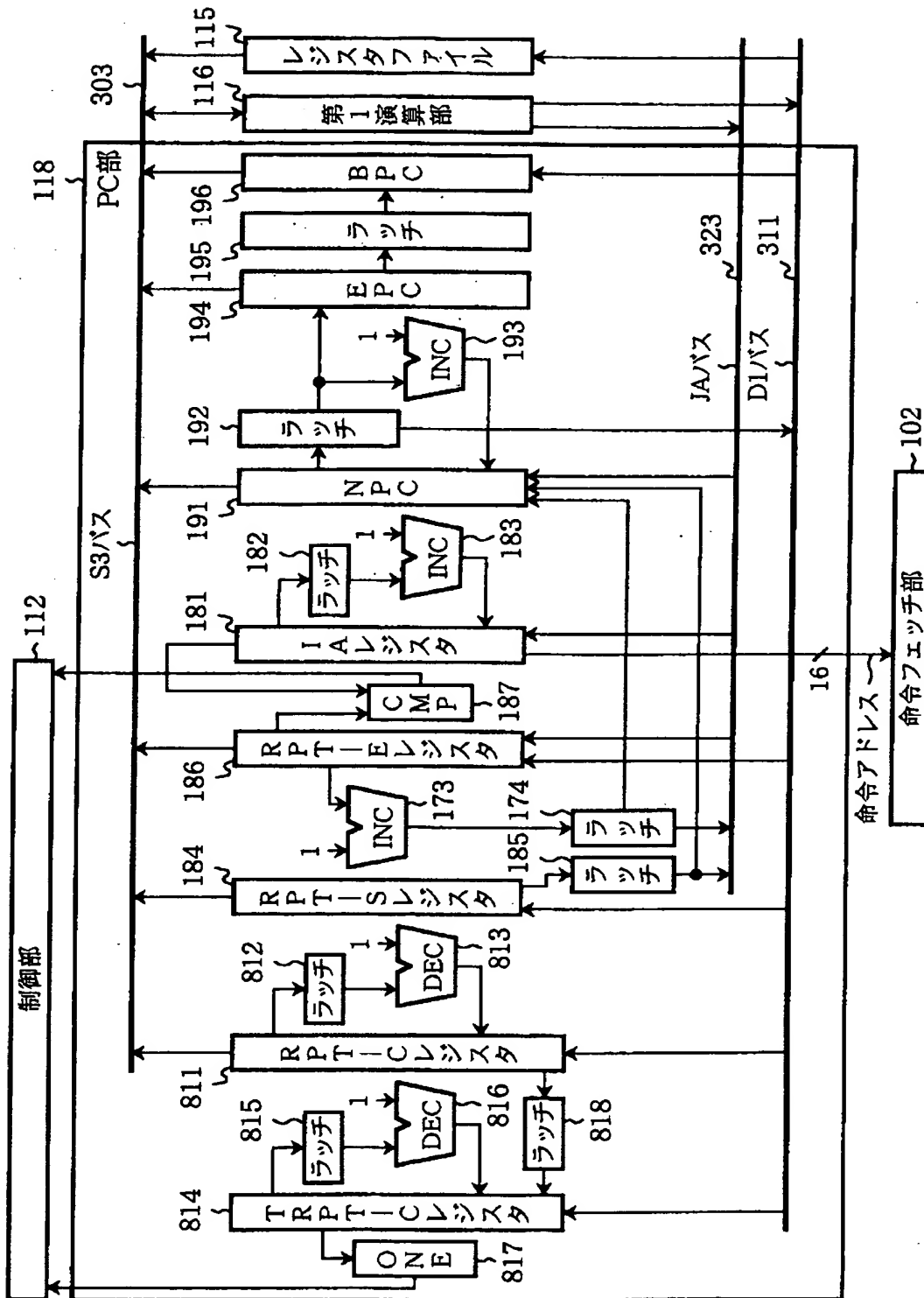


【図 2 9】



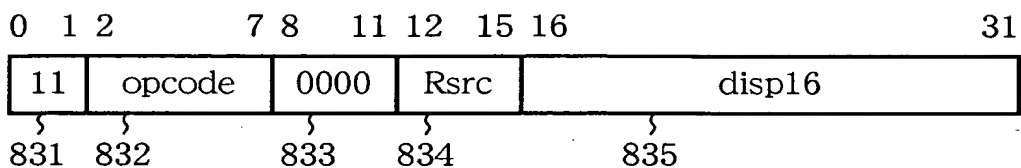


【図 30】



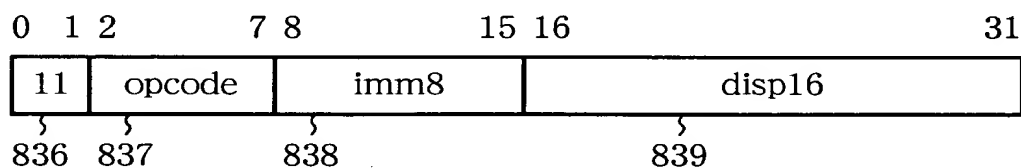
【图 3 1】

ビット番号 DREP Rsrc, disp16



【图 3 2】

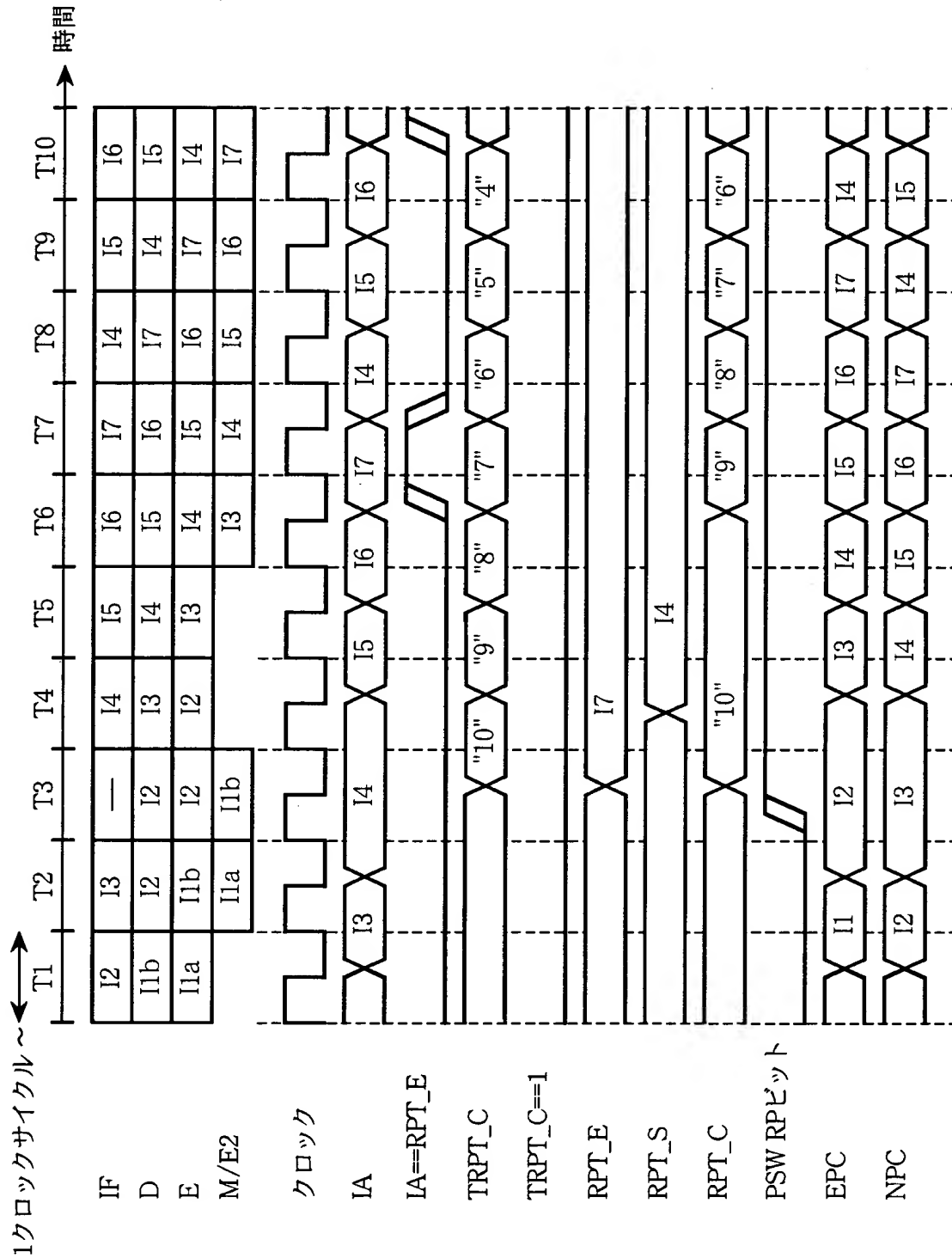
ビット番号 DREPI imm8, disp16



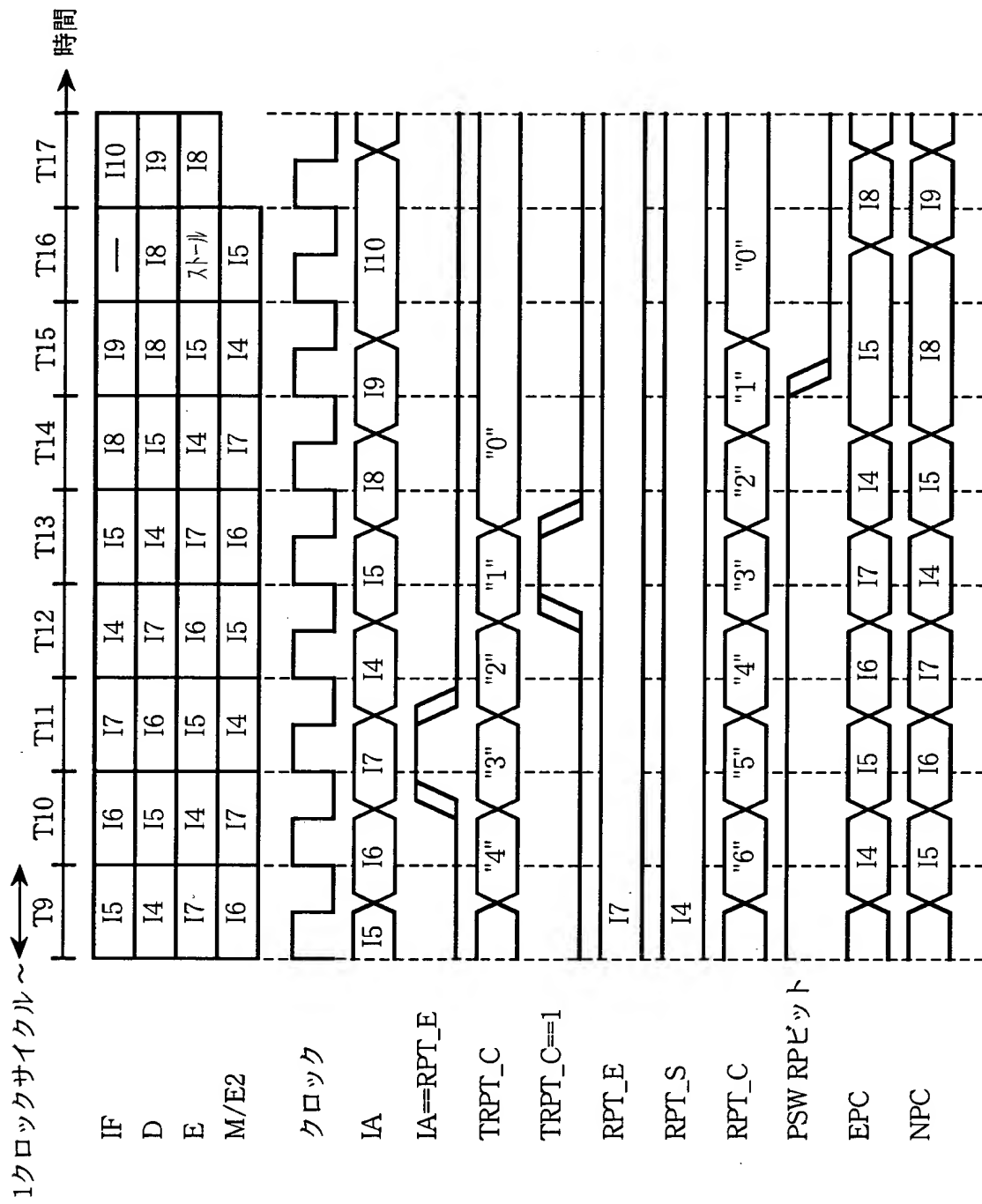
【图 3 3】

LD2W	R0,@R8+				; I1a
LD2W	R4,@R9+				; I1b
DREP	R10,rep_end				; I2
LD2W	R2,@R8+	CLRAC	A0		; I3
rep_start :					
LD2W	R6,@R9+	MAC	A0,R0,R4		; I4
LD2W	R0,@R8+	MAC	A0,R1,R5		; I5
LD2W	R4,@R9+	MAC	A0,R2,R6		; I6
rep_end :					
LD2W	R2,@R8+	MAC	A0,R3,R7		; I7
RACHI	R0,A0,#0	NOP			; I8
:					; I9
:					::

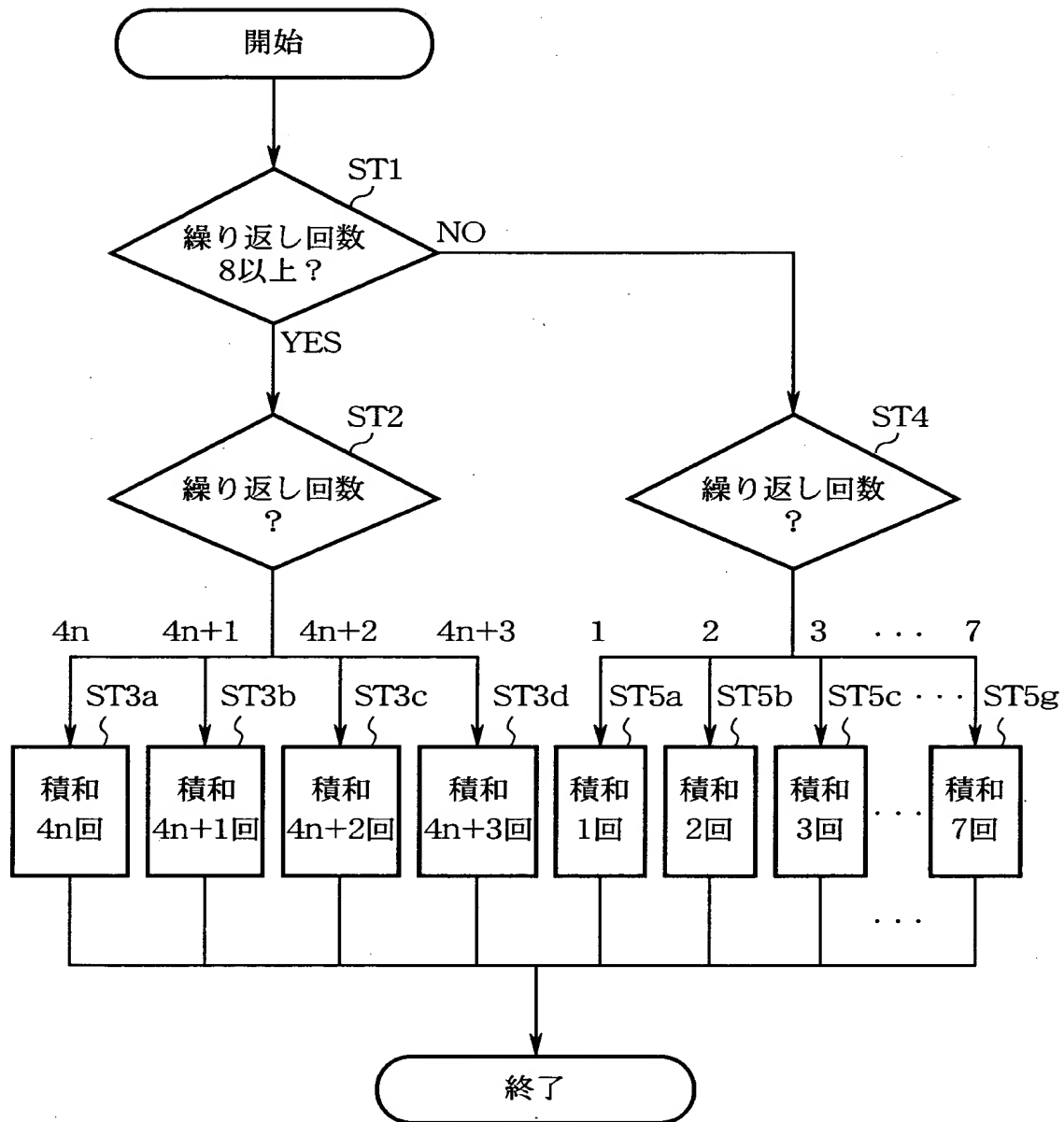
【図 34】



【図 35】



【図 36】



【書類名】 要約書

【要約】

【課題】 繰り返し回数がダイナミックに変化する場合、繰り返し回数を条件判定するためのオーバーヘッドが大きくなり高い性能が得られない等の課題があった。

【解決手段】 比較器 1 7 2 は、R P T \_ B レジスタ 1 7 1 に保持されている値と、I A レジスタ 1 8 1 に保持されている次にフェッチする命令のアドレスとを比較し、これらの値が一致しているか否かを示す一致情報を出力する。制御部 1 1 2 は、リピートブロックの最終の繰り返し処理においてその一致情報に基づきハードウェア的にリピートブロックの次の命令への命令処理シーケンス切り替えを行うための制御信号を生成する。

【選択図】 図 1 0

出 願 人 履 歴 情 報

識別番号 [000006013]

1. 変更年月日 1990年 8月24日  
[変更理由] 新規登録  
住 所 東京都千代田区丸の内2丁目2番3号  
氏 名 三菱電機株式会社